



FACULTAD DE FILOSOFÍA Y LETRAS. DEPARTAMENTO DE GEOGRAFÍA

Programa de DOCTORADO EN CIENCIAS HUMANAS: GEOGRAFÍA,
ANTROPOLOGÍA Y ESTUDIOS DE ÁFRICA Y ASIA

TESIS DOCTORAL

DEEP LEARNING Y BIG DATA EN CARTOGRAFÍA DIGITAL

Creación de inteligencias artificiales para el
tratamiento de ortofotografías y sistemas de
información geográfica tridimensionales

Memoria para optar al grado de Doctor en Geografía presentada por:
DAVID MORILLAS ARMENDÁRIZ

Este trabajo ha sido dirigido por:
Dr. JAVIER ESPIAGO GONZÁLEZ
Director del Servicio de Cartografía
Universidad Autónoma de Madrid

Madrid, Mayo 2021

Agradecimientos

La presente tesis ha sido el resultado de más de un lustro de investigación bibliográfica y desarrollo de nuevas herramientas cartográficas, lo cual no hubiese sido posible sin el apoyo y la paciencia tanto de mi familia como de mis amigos; gracias a todos ellos los resultados de casi una década dedicada al Big Data y a la Inteligencia Artificial, tanto a nivel profesional como académico, se plasman en este documento en el cual se aplican los últimos avances existentes en estos campos a la cartografía digital.

Me gustaría asimismo agradecer el apoyo que he recibido desde el principio de esta aventura del Servicio de Cartografía de la Universidad Autónoma de Madrid (SCUAM), tanto por parte de mis compañeros como en especial de mi director de tesis, el Doctor Javier Espiago, y de Carlos Almonacid, quienes me transmitieron su pasión por la cartografía, me dieron la oportunidad de unirme al SCUAM y cuyos consejos siempre me han ayudado a avanzar. Por último también quisiera hacer una mención especial de la Doctora Rosa Cañada, mi tutora en este proyecto, quien siempre me ha animado y ayudado a gestionar todos los trámites administrativos que implica la realización de un doctorado.

Resumen

Como consecuencia del desarrollo tecnológico experimentado en los últimos años, actualmente nos encontramos ante una avalancha de datos, los cuales ofrecen oportunidades en el campo de la geografía hasta ahora inimaginables. Este maremágnum de datos supone a la vez un reto y una gran oportunidad de acceso al conocimiento, para lo que se hace necesaria la adaptación de estrategias y técnicas basadas en la Inteligencia Artificial y el Big Data al trabajo con datos geográficos.

En esta investigación se comprueba la aplicabilidad en el campo de la cartografía digital de distintas técnicas de procesamiento de datos que actualmente son utilizadas principalmente para la corrección, manipulación, enriquecimiento de grandes volúmenes de datos y su correcta visualización.

Las técnicas aquí analizadas no han sido aplicadas al campo de la información geográfica hasta ahora, al menos de forma general, sin embargo los resultados de esta investigación muestran cómo no sólo es factible su aplicación sino que, de hecho, ésta mejora y agiliza en gran medida distintos procedimientos de manipulación de datos geográficos. Dentro de las técnicas analizadas cabe destacar aquellas que se basan en la Inteligencia Artificial, mediante la aplicación de Deep Learning.

Dentro de la aplicabilidad de diversas redes neuronales existentes para el tratamiento de imágenes, se ha podido comprobar que si bien en determinados casos ofrecen unos resultados que superan a los proporcionados por las técnicas utilizadas hasta ahora, en el análisis y tratamiento de datos geográficos, estos resultados pueden mejorarse mucho entrenando nuevos modelos de redes neuronales profundas con datos geográficos, basadas en arquitecturas

aptas para la realización de las tareas buscadas. En el caso de las inteligencias artificiales generadas en esta investigación se ha recurrido a una arquitectura cGAN (Redes Generativas Antagónicas Condicionales), por lo tanto, disponiendo de arquitecturas adecuadas y entrenando las redes neuronales con los datos convenientes (como ortoimágenes debidamente seleccionadas y tratadas), pueden automatizarse una amplia variedad de trabajos de procesamiento de la información geográfica que hasta ahora debían hacerse manualmente, o al menos necesitaban de intervención humana (como por ejemplo la generación de mapas de usos del suelo o la corrección de errores), además de la automatización de procesos, las inteligencias artificiales orientadas al trabajo con datos geográficos que se han creado permiten el enriquecimiento de los datos y la mejora de la calidad de los mismos, no sólo corrigiendo errores sino agregando información faltante con los datos más plausibles basados en el aprendizaje de estas redes, ya sea completando la información oculta por nubosidad, censura o deterioro de los datos (lo cual a su vez implica posibles aplicaciones militares de ocultación de zonas sensibles de forma mucho más verosímil que las técnicas actuales), consiguiendo incluso restaurar ortoimágenes antiguas mejorando su resolución y asignándoles colores realistas, alcanzando éstas una calidad y características similares a las ortofotografías más actuales con las cuales han sido entrenadas.

Por otra parte, esta investigación explora la posibilidad de aplicación de técnicas de tratamiento de datos y visualización de los mismos para extraer conocimiento a partir de la información disponible en diversas fuentes, para ello se han desarrollado dos herramientas orientadas al tratamiento y a la visualización de datos geográficos. La primera de estas herramientas consiste en la integración de distintas fuentes de ortofotografía digital, mapas vectoriales e información contenida en diversas bases de datos geográficas, proyectables sobre modelos de elevación del terreno globales en un sistema de información geográfica con realidad virtual, que permite no sólo el acceso a la información presente en las distintas bases de datos haciendo clic en el elemento a consultar que se está visualizando, sino que además permite el tratamiento de datos y la edición gráfica a tiempo real desde el propio sistema de información geográfica,

sin la necesidad de descarga, ya que la herramienta se encuentra disponible a través de internet y es consultable a través de los navegadores web. La segunda de las herramientas desarrolladas es un software de tratamiento de datos tridimensionales, esta herramienta agiliza el tratamiento y la visualización de este tipo de datos, pudiendo ingestarse éstos en distintos formatos. Mediante el uso de esta herramienta pueden realizarse correcciones y filtros a los datos procedentes, por ejemplo, de nubes de puntos, lo cual permite al usuario generar salidas gráficas con mayor resolución y corrección de errores que las disponibles actualmente, gracias a distintos métodos de interpolación, pudiendo de esta forma extraer conocimiento a partir de los datos en bruto de forma sencilla gracias a la interfaz gráfica que se ha creado y a la automatización de los procesos que se ha incluido en este software.

Gracias a la investigación desarrollada en esta tesis han podido crearse las primeras inteligencias artificiales basadas en Redes Generativas Antagónicas Condicionales capaces de automatizar diferentes labores geográficas y realizar tratamientos a ortoimágenes que hasta ahora eran imposibles, además de crear herramientas para el tratamiento y visualización de datos bidimensionales y tridimensionales que automatizan, agilizan y facilitan algunos de los procesos más tediosos en el trabajo con información geográfica, devolviendo datos más precisos y de mayor resolución que los existentes hasta este momento.

Abstract

As a result of recent technological developments, there is currently an overwhelming amount of data, which bring new opportunities to the field of geography. This mare magnum of data entails both a challenge and an opportunity to generate new knowledge, which may require the adaptation of strategies and techniques based on Artificial Intelligence and Big Data to the work with geographic data.

This thesis tests the applicability to the field of digital cartography of different techniques of data processing that are currently used mainly for correction, manipulation, enrichening of large volumes of data, and their correct visualisation.

The techniques analysed in this thesis have not been applied to the field of geographic information up to this date, at least widely. However, the results of this research show how the application of these techniques is not only feasible, but also it enhances and speeds up different procedures of geographic data manipulation. Among the analysed techniques it is relevant to highlight those based on Artificial intelligence be means of the use of Deep Learning.

Regarding the applicability of several existing neural networks for image processing, it was found that, despite these networks outperform the most frequently used techniques, the results can be greatly enhanced for the analysis and processing of geographic data if new models of deep neural networks are trained with this kind of data, based on architectures appropriate for the needed tasks. Regarding the artificial intelligences generated in this research, a cGAN architecture (Conditional Generative Adversarial Networks) was used, thus using adequate architectures and training the neural networks with appropriate data (such as orthoimages duly selected and processed). This allows the automation of a wide variety of geographic data processing tasks that, up to this date, had to be done manually or at least with human intervention (such as the generation of

Land Use Land Cover (LULC) maps or the correction of errors), as well as the automation of several processes. The artificial intelligences developed in this research allow enriching and enhancing the quality of the data, not only correcting errors but also adding missing information with the most plausible data based on the learning of the networks, either completing the information hidden by clouds, censoring, or data deterioration (which in turn also involves potential military applications around the concealment of sensitive zones in much more plausible ways than the currently used techniques). These techniques are even capable of restoring old orthoimages, enhancing their resolution and applying realistic colouring, which results in images with similar quality and characteristics than the more current orthophotographs the networks have been trained with.

Moreover, this research explores the use of techniques of data processing and visualisation to extract knowledge from the information available in different sources. To do so, two tools oriented to the processing and visualisation of geographic data have been developed during the research. The first of these consists of the integration of different sources of digital orthophotography, vector maps and information contained in different geographic databases, projected onto global digital elevation models (DEMs) in a virtual reality geographic information system. This allows not only accessing to the information available in those different databases by just clicking on the element to explore that is being visualised, but also processing the data and editing it graphically in real time from the very geographic information system. Since the tool is available online and can be consulted using web browsers, all this can be done without the need to download the data. The second of the developed tools is a 3D data processing software that speeds up the processing and visualisation of this kind of data, being able to feed the data in different formats. By using this tool, corrections and filters can be applied to data coming from, for instance, point clouds, which allows the user to generate graphical outputs with higher resolution and greater accuracy than by using the previously available tools. This is possible through different interpolation methods which make it possible to easily extract information and generate new knowledge from the raw data due to the

graphic interface that has been created and to the automation of the processes included in the software.

Thanks to the research developed in this thesis, the first artificial intelligences based on Conditional Generative Adversarial Networks able to automate different geographic tasks and process orthoimages in a way that was impossible until now have been created, as well as tools for the processing and visualisation of 2D and 3D data that automate, speed up and facilitate some of the most cumbersome processes in the work with geographic information, returning data that are more accurate and with higher resolution than the previously existing tools.

Índice

1. Introducción	21
1.1. Objetivos	24
1.2. Hipótesis.....	26
1.3. Motivación	30
1.4. Contribuciones.....	32
1.5. Estructura de la tesis	40
1.6. La problemática de las ortoimágenes antiguas	43
1.7. La creación automática de mapas de usos del suelo.....	45
1.8. La inteligencia artificial.....	49
2. Inteligencias Artificiales Cartográficas.....	57
2.1. Entrenamiento de los modelos	60
2.2. Metodología seguida para el entrenamiento de los modelos propios.....	62
2.3. La arquitectura de los modelos.....	83
2.3.1. Generador.....	88
2.3.2. Discriminador	99
2.3.3. Entrenamiento.....	108
2.3.4. Evaluación del entrenamiento	119
2.4. Librerías utilizadas.....	121
2.5. Hardware utilizado en los entrenamientos	122
2.6. Metodología de trabajo para la aplicación práctica de redes neuronales en la cartografía	123
2.6.1. Introducción de datos en Sistema de Información Geográfica y generación de teselas	126
3. El Color.....	147
3.1. Adaptación de técnicas y modelos preexistentes para su aplicación en la coloración de ortoimágenes aéreas.....	148
3.2. Antecedentes	150
3.3. Transferencia de color entre imágenes.....	152
3.4. Colorización de imágenes mediante Deep Learning	156
3.5. La transferencia de estilo.....	161
3.6. Pruebas con el modelo DeOldify	173

3.6.1. Conclusión de las pruebas con técnicas conocidas.....	175
3.7. Modelos propios	177
3.7.1. DeepMapsColor19	180
3.7.2. DeepMapsColor15	183
3.7.3. DeepMapsColor17	185
3.7.4. DeepMapsColorMultizoom	187
3.8. Conclusiones.....	190
4. Resolución y enfoque	191
4.1. Técnicas clásicas de interpolación para mejorar la resolución.....	193
4.1.1. Nearest Neighbor o vecino más próximo.....	194
4.1.2. Interpolación bilineal	195
4.1.3. Interpolación Bicúbica	197
4.2. Aplicación de Redes Generativas Antagónicas de súper resolución o SRGAN	200
4.3. Comparativa entre los resultados de modelos de Deep Learning existentes y del modelo de creación propia DeepMapsDespixela	202
4.3.1. Resultados obtenidos comparados con la imagen original	203
4.4. Técnicas clásicas para mejorar el enfoque	204
4.4.1. Técnica de sharpening o afilado.....	204
4.4.2. Enmascaramiento no nítido o Unsharp Masking	207
4.5. Modelos propios	208
4.5.1. DeepMapsOptimiza.....	208
4.5.2. DeepMapsEnfocayColoreaHR	210
4.5.3. DeepMapsOptimiza17BN.....	212
4.5.4. DeepMapsOptimiza17	216
4.5.5. DeepMapsDespixela	217
5. Corrección de errores y artefactos	221
5.1. Ruido.....	224
5.1.1. Non-Local Means Denoising	226
5.2. Inpainting o relleno de la imagen	229

5.2.1. Ecuación biarmónica	233
5.2.2. Método de marcha rápida	239
5.2.3. Convoluciones parciales con una actualización automática de la máscara	241
5.3. Modelos propios	246
5.3.1. DeepMapsRuido	249
5.3.2. DeepMapsRuido y Desenfoque.....	251
5.3.3. DeepMapsRuido y DesenfoqueBN.....	253
5.3.4. DeepMapsInpainting	254
5.3.5. DeepMapsNubes	257
5.3.6. DeepMapsReparaHipsométrico	259
5.3.7. DeepMapsRestaura15	260
5.3.8. DeepMapsRestaura17	262
5.3.9. DeepMapsRestaura19	263
6. Generación automatizada de mapas de usos del suelo.....	265
6.1. La adaptación de dominios	276
6.2. Modelos propios	284
6.2.1. DeepMapsBordes	284
6.2.2. DeepMapsBordes15	286
6.2.3. DeepMapsBordes17	287
6.2.4. DeepMapsHillshades	291
6.3. La segmentación semántica	294
6.4. Modelos propios	297
6.4.1. DeepMapsUsosNatura	297
6.4.2. DeepMapsBosques.....	301
6.4.3. DeepMapsUsosSIOSE.....	304
6.4.4. DeepMapsSuperficies	308
7. Casos de uso de las inteligencias artificiales creadas	311
7.1. Caso 1: Eliminación de parches en el mapa hipsométrico	313

7.2. Caso 2. Eliminación de parches de censura de datos	323
7.2.1. Caso 2. Ejemplo 1 - Recreación de datos correspondientes a las áreas censuradas entre Carabanchel y Pozuelo de Alarcón (Comunidad de Madrid). 323	
7.2.2. Caso 2. Ejemplo 2: Recreación de datos correspondientes al área censurada de la Base de Apoyo de Radiotransmisores	327
7.2.3. Caso 2. Ejemplo 3: Recreación de datos correspondientes al área censurada de la Base de Helicópteros F.A.M.E.T	330
7.3. Caso 3: Eliminación de nubes en las imágenes del LandSat8	336
7.4. Caso 4: Corrección de defectos en las ortoimágenes del Vuelo Americano Serie B (1956-1957)	339
7.4.1. Caso 4. Ejemplo 1: Corrección de anomalía cercana a la Plaza de Toros de las Ventas (Madrid).....	339
7.4.2. Caso 4. Ejemplo 2: Corrección de anomalía en el actual distrito de San Blas (Madrid)	342
7.4.3. Caso 4. Ejemplo 3: Corrección de roturas en ortofotografía de las áreas de Manoteras y Pinar de Chamartín (Madrid)	344
7.4.4. Caso 4. Ejemplo 4: Corrección de arañazo en ortofotografía de área situada entre Fuenlabrada y Leganés (Madrid)	346
7.5. Caso 5: Corrección defectos en las ortoimágenes del vuelo de Ruiz de Alda (1929-1930).....	349
7.5.1. Caso 4. Ejemplo 1: Eliminación de polvo en ortofotografía de área situada al norte de San Vicente del Raspeig (Alicante)	349
7.5.2. Caso 5. Ejemplo 2: Eliminación de polvo y arañazos en ortofotografía de Elche.....	352
7.5.3. Caso 5. Ejemplo 3: Corrección de unión mal realizada entre ortofotografías del Vuelo de Ruiz de Alda en el área de Murcia.....	355
7.6. Caso 6: Mejora de la resolución	359
7.6.1. Caso 6. Ejemplo 1: Campo de cultivo (Comunidad de Madrid)	359
7.6.2. Caso 6. Ejemplo 2. Paso de cebra (Comunidad de Madrid).....	362

7.6.3. Caso 6. Ejemplo 3: Carretera (Comunidad de Madrid)	364
7.6.4. Caso 6. Ejemplo 4: Embalse (Cantabria)	367
7.6.5. Caso 6. Ejemplo 5: Parque del Retiro (Comunidad de Madrid)	370
7.7. Caso 7: Restauración de ortofotografías de la Cuenca del Segura del Vuelo Ruiz de Alda de 1929-1930	373
7.7.1. Caso 7: Ejemplo 1: El Palmar (Murcia).....	375
7.7.2. Caso 7: Ejemplo 2: Campos de labor (Murcia)	377
7.7.3. Caso 7: Ejemplo 3: Plaza de Toros y Estadio de la Condomina (Murcia)	379
7.7.4. Caso 7: Ejemplo 4: Elche (Alicante)	381
7.8. Caso 8: Restauración de ortofotografías de Madrid del Vuelo Americano (Serie B, 1956-1957).....	383
7.8.1. Caso 8: Ejemplo 1: Jardines del Campo del Moro (Madrid).....	383
7.8.2. Caso 8: Ejemplo 2: Plaza de Moncloa (Madrid).....	385
7.8.3. Caso 8: Ejemplo 3: Plaza de Colón (Madrid).....	387
7.8.4. Caso 8: Ejemplo 4: Paseo del Prado (Madrid).....	390
7.8.5. Caso 8: Ejemplo 5: Atocha (Madrid).....	392
7.8.6. Caso 8: Ejemplo 6: Plaza de Manuel Becerra (Madrid)	394
7.8.7. Caso 8: Ejemplo 7: Puerta de Alcalá (Madrid).....	397
7.9. Caso 9 Generación automatizada de mapas de usos del suelo con categorías de SIOSE	399
7.9.1. Caso 9. Ejemplo 1: Mapa de usos del suelo de Algete (Madrid).....	400
7.9.2. Caso 9. Ejemplo 2: Mapa de usos del suelo de Meco (Madrid)	401
7.9.3. Caso 9. Ejemplo 3: Mapa de usos del suelo de la Cuenca del Henares (Madrid)	402
7.10. Caso 10: Mapas de densidad de cobertura de arboles	403
7.10.1. Caso Ejemplo 1: Densidad de árboles en South Western National Park (Tasmania). Zona1.....	405
7.10.2. Caso Ejemplo 2: Densidad de árboles en South Western National Park (Tasmania). Zona2.....	406

7.10.3. Caso Ejemplo 3: Densidad de árboles en South Western National Park (Tasmania). Zona3.....	407
7.10.4. Caso Ejemplo 4: Densidad de árboles en South Western National Park (Tasmania). Zona4.....	408
7.11. Caso 11: Elaboración de mapas de uso de suelo a partir de ortoimágenes satelitales de menor escala	409
7.11.1. Caso 11: Ejemplo 1: Mapa de usos del suelo del área entre Lerma y Aranda de Duero	410
7.11.2. Caso 11: Ejemplo 2: Mapa de usos del suelo de zona sur de Burgos ...	411
7.11.3. Caso 11: Ejemplo 3: Mapa de usos del suelo del área entre Cuéllar y Cantalejo	412
7.12. Caso 12 Generación automatizada de mapas de usos del suelo con categorías de Natura2000	413
7.12.1. Caso 12: Ejemplo 1: Creación de mapa de usos a partir de datos del SIGPAC	414
7.12.2. Caso 12: Ejemplo 2: Generación automática de mapa de usos a partir de datos del PNOA de Máxima Actualidad	415
7.12.3. Caso 12: Ejemplo 3: Generación automática de mapa de usos a partir de datos del PNOA 2018	417
7.12.4. Caso 12: Ejemplo 4: Generación automática de mapa de usos del Monte de Valdelatas a partir de datos del PNOA de Máxima Actualidad	418
7.12.5. Caso 12: Ejemplo 5: Generación automática de mapa de usos a partir de datos de la capa Esri Aéreo del parque South Western National Park	420
8. Elaboración del software 3D	423
8.1. Calidad de los datos	423
8.1.1. Etapas del proceso de Data Quality	424
8.2. Desarrollo del software online.....	430
8.2.1. Lenguajes y librerías	437
8.3. Desarrollo del Software de escritorio “MEL”	438

8.3.1. Lenguajes, librerías y programas de terceros.....	450
8.4. Fuentes de datos.....	451
8.4.1. Fuentes de datos del software de realidad virtual desarrollado para su uso online	451
8.4.2. Capas para el estudio del relieve	458
8.4.3. Capas para el estudio de la geología	459
8.4.4. Ortofotos Históricas.....	464
8.4.5. Capas para teledetección atmosférica	467
8.5. Fuentes de datos del software MEL.....	468
8.5.1. Otras fuentes de datos	469
9. Metodología de trabajo con datos 3D	471
9.1. Descripción de la GUI (interfaz gráfica de usuario) de MEL y metodología de trabajo general	474
9.1.1. Herramientas avanzadas incluidas en MEL.....	478
9.2. Generación automatizada de imágenes individuales a partir de datos LiDAR	488
9.3. Elaboración masiva de imágenes a partir de datos LiDAR, para el análisis exploratorio y la detección de estructuras.....	501
9.4. Análisis del relieve mediante realidad virtual y tratamiento dinámico de los datos sin necesidad de descarga	515
10. Casos de uso del visor VR y MEL	527
10.1. Valle de Ucanca (Santa Cruz de Tenerife).....	530
10.2. Presa y cascadas de Ézaro (La Coruña)	535
10.3. Raña de Anchuras (Toledo).....	540
10.4. Karst en yesos de Sorbas (Almería)	543
10.5. Edificio del Teide (Santa Cruz de Tenerife)	546
10.6. Edificios de la pared de Las Cañadas (Tenerife)	550
10.7. Roque Nublo (Tenerife)	557
10.8. Aplicaciones geoarqueológicas	560
10.8.1. Lugar Porto Outeiro.....	563
11. Conclusiones	579
12. Anexos	595
12.1. Anexo I: Evolución de las inteligencias artificiales creadas	595

12.1.1. Ejemplo del proceso de entrenamiento del modelo DeepMapsColor19.	595
12.1.2. Ejemplo del proceso de entrenamiento del modelo DeepMapsInpainting	
.....	600
12.1.3. Ejemplo del proceso de entrenamiento del modelo DeepMapsBosques	601
12.1.4. Ejemplo del proceso de entrenamiento del modelo DeepMapsBordes17	
.....	603
12.2. Anexo II: Hillshades multiparámetro y mapas hipsométricos multialtitud	604
12.3. Anexo III: Generación automatizada de mapas hipsométricos.....	610
12.3.1. Santa Cruz de la Palma (Las Palmas de Gran Canaria).....	610
12.3.2. Comunidad de Madrid	612
13. Bibliografía	613

1. Introducción

La amplia difusión, tanto de la georreferenciación como de herramientas de visualización de información geográfica, ha permitido la integración de datos de muy diversas fuentes. Esto, unido a los constantes desarrollos tecnológicos, está permitiendo que se tome consciencia, a nivel general, de la importancia de la geovisualización y el potencial de esta para contribuir al estudio de los diferentes fenómenos, tanto humanos como naturales, que tienen lugar en nuestro planeta. Al mismo tiempo, ha puesto de manifiesto el reto que supone el almacenamiento, el procesamiento y la visualización de los datos, así como la transformación de los datos brutos en información y conocimiento aprovechable.

A lo largo de esta tesis vamos a intentar esclarecer el estado actual de diversas tecnologías de última generación, las cuales podrían tener una aplicación directa para la resolución de distintas problemáticas existentes en la cartografía, así como su integración para una aplicación práctica en la cartografía digital. Estas tecnologías son principalmente el Big Data y la Inteligencia Artificial, con sus distintas modalidades. Gracias a las conclusiones obtenidas durante este estudio, se propondrán distintas metodologías que resuelvan algunos de los problemas existentes, basadas en el conocimiento adquirido durante el desarrollo de este trabajo y en el software desarrollado durante el mismo.

Antes de continuar, debe aclararse que la terminología utilizada en ocasiones no ha sido traducida al castellano, debido no solo a que determinados anglicismos son de uso generalizado en las publicaciones técnicas de la temática aquí tratada, sino principalmente a que el significado de estos préstamos lingüísticos expresa de forma más correcta el concepto que se pretende transmitir, no obstante, en los casos en los que la traducción al castellano no producía pérdidas de significado y eran términos que ya han sido

previamente utilizados en castellano en otras publicaciones, éstos han sido expresados en nuestro idioma.

Algunos de los problemas fundamentales a los que vamos a enfrentarnos serían la posibilidad de restaurar y dar colores realistas, y coherentes con el entorno, a imágenes aéreas y satelitales antiguas obtenidas con tecnologías ya obsoletas.

Por otra parte, una problemática importante que nos encontramos con los datos de reciente adquisición obtenidos utilizando los sistemas actuales, como podrían ser los sensores LiDAR (Light Amplification by Stimulated Emission of Radiation), es la existencia de datos erróneos y anomalías o la propia falta de datos en determinadas áreas, lo cual puede solucionarse con distintos procedimientos que han sido implementados dentro de la herramienta “Morillas Easy LiDAR” (MEL), desarrollada durante esta investigación, y que serán explicados más adelante en los capítulos 8 y 9 y con algunas de las inteligencias artificiales desarrolladas como DeepMapsInpainting, la cual genera datos verosímiles para completar las zonas donde existe ausencia de datos en las ortoimágenes. Los detalles de los entrenamientos de esta y otras inteligencias artificiales, que permiten mejorar los datos de las ortoimágenes, vienen recogidos en los capítulos del 3 al 7.

Por último, un problema existente, tanto en los datos antiguos como en los datos actuales, es la resolución de las imágenes, su enfoque y la existencia de errores como ruido y artefactos, pudiendo todo ello corregirse mediante el uso de distintas técnicas, las cuales son analizadas junto con sus correspondientes resultados en los capítulos 4 y 5.

La mejora de la resolución y la corrección de datos erróneos deben plantearse desde dos perspectivas diferentes dependiendo del tipo de datos que analicemos, siendo necesario adoptar unas estrategias distintas para los datos

en dos dimensiones que las utilizadas para los datos tridimensionales. Es por ello por lo se enfoca este problema de forma independiente, pudiendo resolverse, en el caso de los datos en dos dimensiones, mediante el uso de las inteligencias artificiales creadas durante este trabajo, que se exponen en los capítulos 4 y 5, y en el caso de los datos tridimensionales, mediante la utilización del software MEL, cuya elaboración y utilización se expone en los capítulos 8 y 9.

Cabe mencionar que para la obtención de los resultados deseados han debido crearse nuevas inteligencias artificiales adaptadas específicamente para su uso en cartografía, mediante técnicas de Deep Learning, basadas en la arquitectura de las redes generativas antagónicas condicionales, con un generador de tipo U-Net y un discriminador de tipo PatchGAN.

En el presente documento se proponen diversas metodologías de recepción y procesamiento de la información geográfica, orientadas principalmente a las labores cartográficas de tratamiento y generación de datos enfocados a la teledetección y el análisis de geomorfologías. Como parte del estudio realizado, se han desarrollado diversas herramientas que facilitan esta labor, entre las cuales se incluyen varias inteligencias artificiales capaces de realizar labores cartográficas de tratamiento de ortoimágenes y generación de mapas de manera automatizada, además de un software libre para el tratamiento y análisis de datos LiDAR y un SIG (Sistema de Información Geográfica) de realidad virtual online.

1.1. Objetivos

El objetivo de este documento es presentar diversas contribuciones a las técnicas y metodologías aplicadas a la cartografía digital mediante la aplicación de la Inteligencia Artificial y técnicas propias del Big Data, así como el desarrollo de herramientas que puedan facilitar las labores cartográficas y de teledetección, tanto en lo relacionado con datos bidimensionales como con datos tridimensionales. Todas estas contribuciones hacen hincapié en la mejora y el enriquecimiento de los datos, así como en su manejo, con el objetivo de extraer información útil.

Como consecuencia de las pruebas realizadas, se ha comprobado que, si bien algunas de las técnicas existentes para el tratamiento de imágenes son aplicables a la cartografía, los resultados obtenidos en el caso de las redes neuronales cuando los modelos son entrenados específicamente con datos procedentes de ortoimágenes aéreas y satelitales, aportan mejores resultados que los algoritmos de procesamiento de imágenes y las redes neuronales existentes entrenadas con imágenes de otro tipo. En este sentido se han entrenado varios modelos de redes neuronales con este tipo de datos para poder sentar las bases de la aplicación de la Inteligencia Artificial a la cartografía.

La creación de estos modelos tiene como meta la demostración de la validez de las hipótesis planteadas en relación con la aplicación de la Inteligencia Artificial y técnicas propias del procesamiento de datos utilizadas en el Big Data, a la cartografía y la exploración de las posibilidades que ofrecen estas tecnologías, así como de la validez de las bases planteadas para el trabajo en este campo.

Las conclusiones y resultados obtenidos son el fruto no sólo de pruebas realizadas con éxito, sino también de los inconvenientes surgidos durante las mismas, los cuales han permitido comprobar qué estrategias son más

adecuadas a la hora de trabajar con redes neuronales en cartografía y que algoritmos resultan más eficientes para el tratamiento masivo de datos, así como para el manejo de éstos. Los conocimientos extraídos han permitido, por lo tanto, generar inteligencias artificiales pioneras en algunas tareas y que superan a las existentes al ser aplicadas a datos cartográficos, además de herramientas que permiten la agilización del trabajo y la mejora de los resultados al procesar datos LiDAR o realizar labores de teledetección con capas servidas a través de internet.

A pesar de que el aprendizaje es un concepto muy general, en lo que se refiere a este trabajo se puede definir como la tarea que tiene como objetivo encontrar una función que asigne a una entrada (por ejemplo, una ortoimagen aérea) una salida (por ejemplo, un mapa de usos del suelo). Más específicamente, la parte de esta investigación dedicada a la aplicación de la Inteligencia Artificial en la cartografía se centra en abordar este tipo de tareas mediante el aprendizaje automático de varias transformaciones no lineales de los datos que se estructuran en capas. Dado que el número de capas en las que se estructuran las redes creadas es alto, inteligencias artificiales creadas en esta tesis pueden ser englobadas dentro del Deep Learning o aprendizaje profundo. Gracias a los modelos entrenados en el desarrollo del presente documento y los resultados obtenidos con ellos se demuestra que el Deep Learning se puede utilizar para resolver distintos problemas y desafíos existentes en el campo de la cartografía digital, como la automatización de procesos hasta ahora manuales que consumen gran cantidad de recursos técnicos y humanos.

1.2. Hipótesis

Las hipótesis planteadas pueden dividirse en dos grupos; en primer lugar las derivadas de la aplicabilidad de la Inteligencia Artificial a la resolución de distintas cuestiones cartográficas y, en segundo lugar, tenemos aquellas hipótesis centradas en la posibilidad tanto de simplificación como de mejora del software existente para el trabajo con datos LiDAR y el procesamiento a través del navegador de datos masivos, tanto bidimensionales como tridimensionales, incluyendo información ráster y vectorial.

En primer lugar, las relativas a la Inteligencia Artificial son las siguientes:

- Pueden mejorarse los resultados de algoritmos clásicos para el tratamiento de imágenes en cartografía aplicando redes neuronales entrenadas con datos de ortoimágenes aéreas y satelitales.
- La aplicabilidad de las redes neuronales en la cartografía depende, en gran medida, del dominio de los datos con los que fueron entrenadas.
- Se pueden adaptar diferentes dominios de las imágenes mediante el uso de redes neuronales entrenadas para tal fin.
- El uso de redes generativas antagónicas condicionales, correctamente entrenadas, permite corregir errores en los datos de forma automática debido a la recreación de los datos que realizan estas redes.
- La corrección de errores en los datos resulta más efectiva cuanto más simplificada esté la imagen de entrada que se le aporta a las redes para la generación de nuevas imágenes; sin embargo, existen límites a esta simplificación, los cuales, si se sobrepasan, provocan que las imágenes

generadas, aunque realistas, no se correspondan con los elementos reales que se pretende recrear.

- Es posible generar mapas de ocupación del suelo utilizando redes neuronales entrenadas con los datos adecuados.
- La selección adecuada de imágenes para el entrenamiento de redes neuronales tiene un efecto directo en el rendimiento de estas.
- Es posible aplicar los modelos entrenados con los datos de un área concreta, siempre y cuando dicha área contenga los elementos suficientes, a otras áreas diferentes.
- La inclusión de ruido, desenfoque o variaciones en el contraste y la luminosidad, mediante el preprocesamiento de las ortoimágenes de entrenamiento de las redes neuronales proporciona mayor versatilidad y permite un entrenamiento más rápido, a la vez que reduce el número de imágenes originales necesarias para un entrenamiento efectivo.
- El ruido, el desenfoque y variaciones del contraste y la luminosidad, cuando son incluidos mediante el preprocesamiento de los datos de entrenamiento permiten a las redes generativas antagónicas condicionales aprender a corregir defectos existentes en los datos originales, considerando no sólo la corrección de los factores de distorsión incluidos mediante preprocesamiento en los datos de entrenamiento, sino también la corrección de otros posibles fallos como los desperfectos en el soporte físico de las ortofotografías o la presencia de elementos extraños.
- El entrenamiento con datos masivos (Big Data) durante las suficientes iteraciones permite a las redes generativas antagónicas condicionales

desarrollar de forma espontánea comportamientos emergentes, los cuales en muchas ocasiones resultan útiles, como por ejemplo la realización de correcciones para las que no habían sido entrenadas específicamente.

- Algunos de los comportamientos surgidos gracias al fenómeno de autoemergencia en sistemas complejos resultan útiles, pero otros suponen un inconveniente (pueden llegar a niveles de abstracción que dificultan la percepción humana de la información contenida en los datos), por lo que debe vigilarse el desarrollo de las inteligencias artificiales, comprobando los resultados que devuelven a medida que avanza su entrenamiento.
- Aplicar un procesamiento previo de los datos de entrenamiento para generar variaciones de éstos, basadas en distintas combinaciones de los parámetros de luminosidad y contraste, permite una mejora muy significativa en la capacidad de las redes neuronales para procesar los datos de ortoimágenes antiguas.
- La generación de variantes de los datos originales mediante el preprocesamiento de éstos permite reducir el número de ortoimágenes originales necesarias para generar un dataset adecuado para el entrenamiento, ya que el número de imágenes disponibles para el entrenamiento se multiplica debido a las distintas variantes obtenidas de los datos originales.
- La simplificación de los datos en el proceso de entrenamiento, mediante el preprocesado de los mismos para generar imágenes de los límites entre unidades, permite utilizar la técnica de adaptación de dominios, con la cual pueden aplicarse modelos entrenados con datos de diferentes fuentes y por lo tanto extender la aplicabilidad de las redes neuronales

cartográficas entrenadas para su uso con un dominio específico a otras fuentes de datos.

En segundo lugar, las relacionadas con el tratamiento de datos LiDAR son las mostradas a continuación:

- Es posible generar un modelo digital del terreno de mayor resolución a los ofrecidos por el IGN (MDT05 y MDT02), a partir de los datos obtenidos por los vuelos LiDAR del Plan Nacional de Ortofotografía Aérea.
- La utilización de técnicas basadas en el modelado tridimensional, como la generación de redes de triángulos irregulares o TIN, mejora en gran medida los resultados obtenidos aplicando, únicamente, los métodos de interpolación a la hora de generar modelos digitales del terreno a partir de nubes de puntos.
- Asimismo, también se puede desarrollar un software libre que facilite el análisis del relieve mediante la automatización de la generación de archivos de forma masiva, derivados de los ficheros LiDAR ofrecidos por el IGN, con diferentes parámetros de altitud, azimuth y factor z, etc.
- El software desarrollado es adecuado para el estudio del relieve, la geomorfología regional y la identificación de estructuras de origen antrópico.
- Es posible desarrollar un SIG online tridimensional con realidad virtual que, además, permita realizar el tratamiento de las imágenes que componen las distintas capas incluidas en el software a tiempo real mientras se utiliza el visor, lo cual posibilita realizar las tareas de procesamiento de imágenes propias de la teledetección de forma online.

1.3. Motivación

Muchos problemas, en el procesamiento de imágenes, gráficos y visión artificial, implican traducir una imagen de entrada en una imagen de salida. Estos problemas, a menudo, se tratan con algoritmos específicos según el objetivo, a pesar de que la tarea es siempre la misma, asignar píxeles a píxeles. Las redes generativas antagónicas condicionales son una solución de propósito general que parece funcionar bien en una amplia variedad de estos problemas.

Durante los próximos capítulos se muestran los resultados obtenidos mediante la aplicación de las inteligencias artificiales entrenadas, con una arquitectura cGAN o de Redes Generativas Antagónicas Condicionales conocida como image-to-image o Pix2Pix, para la realización de varias tareas de importancia en la cartografía digital. En cada caso se utiliza la misma arquitectura para la resolución de las diferentes tareas, simplemente se entrenan los distintos modelos con datos diferentes en base a la función que deben cumplir.

Se ha podido comprobar cómo, a pesar de que algunos de los algoritmos y redes neuronales existentes, cuyo desarrollo está orientado al tratamiento de imágenes, pueden adaptarse para el desempeño de labores cartográficas. Los resultados obtenidos podrían mejorarse si las redes neuronales son entrenadas específicamente con datos de ortoimágenes aéreas y satelitales. Debido a esto, se han entrenado durante varios meses algunos modelos de redes neuronales para su aplicación específica en labores cartográficas. Para obtener los resultados que se muestran a lo largo de este documento, ha sido necesaria una profunda revisión bibliográfica y la realización de numerosas pruebas, hasta llegar a establecer determinados principios que permiten sentar las bases de la aplicación de técnicas propias del Big Data y de la Inteligencia Artificial en el procesamiento de datos, tanto bidimensionales como tridimensionales, en la cartografía digital.

Muchas imágenes presentan problemas de procesamiento cómo son la restauración de estas, su coloración y clasificación semántica (mapas de usos del suelo, densidad forestal, etc.), los cuáles pueden ser considerados como un trabajo de traducción de imagen a imagen; estos problemas y, sobre todo, su resolución de forma automatizada sin necesidad de intervención humana, suponen algunos de los mayores retos a los que se enfrenta la cartografía actualmente. Por ello, a lo largo de este documento se van a resolver dichos problemas con la aplicación de las últimas tecnologías en algoritmos de tratamiento de imagen, modelado tridimensional, redes neuronales y Deep Learning, comparando los resultados obtenidos con los conseguidos mediante el uso de técnicas más tradicionales de procesamiento de imágenes y datos tridimensionales.

El propósito final de esta investigación es descubrir las posibilidades de aplicación de la Inteligencia Artificial a la cartografía (como por ejemplo la recuperación de imágenes antiguas dañadas, la coloración de imágenes en escala de grises y la generación de mapas temáticos, como pueden ser los mapas de ocupación de suelos a partir de ortoimágenes aéreas y satelitales, de forma automatizada), así como desarrollar software y metodologías que permitan tanto la teledetección, el estudio de geomorfologías de forma más detallada que las existentes actualmente y la identificación de estructuras de origen antrópico, aprovechando al máximo la resolución de los datos contenidos en las nubes de puntos obtenidas de diversas fuentes, tanto nacionales (como el Instituto Geográfico Nacional, IGN) como internacionales (como el Servicio Geológico de Estados Unidos, USGS) y el establecimiento de determinados principios a tener en cuenta a la hora de aplicar el Big Data y la Inteligencia Artificial al trabajo cartográfico.

1.4. Contribuciones

El Deep Learning o aprendizaje profundo implica muchos tipos de algoritmos y aplicaciones diferentes; por otra parte, los límites entre lo que es Deep Learning y lo que no lo es, no están bien definidos. La mayoría de los investigadores de la comunidad científica dedicados al Deep Learning están especializados en temas específicos que se aplican a sus áreas de interés. Al igual que ellos, el objetivo principal de este documento es compartir varias contribuciones sobre diferentes factores clave que pueden desempeñar un papel fundamental en la aplicación de métodos de Deep Learning a la cartografía digital y crear herramientas que faciliten el tratamiento y la visualización masiva de datos.



Figura 1.1. Restauración de ortofotografía aérea del Vuelo Interministerial (1973-1986) aplicando uno de los modelos de creación propia (DeepMapsRestaura17)

Fuente: elaboración propia, a partir de datos del IGN

La contribución principal presentada en este documento es crear una serie de inteligencias artificiales y algoritmos que realicen de forma automática labores de mejora de la calidad de los datos (corrección de errores y restauración de ortoimágenes aéreas) y enriquecimiento de los mismos (generar colores y mejorar la calidad de ortoimágenes aéreas antiguas, generación de mapas de ocupación del suelo), así como el establecimiento de las bases de la

aplicabilidad de la inteligencia artificial en el manejo de datos cartográficos (capítulos 2 a 7) y la creación de herramientas para la consulta y manipulación de los mismos, las cuales permiten al usuario el máximo aprovechamiento de la información que se le brinda (capítulos 8 a 10).

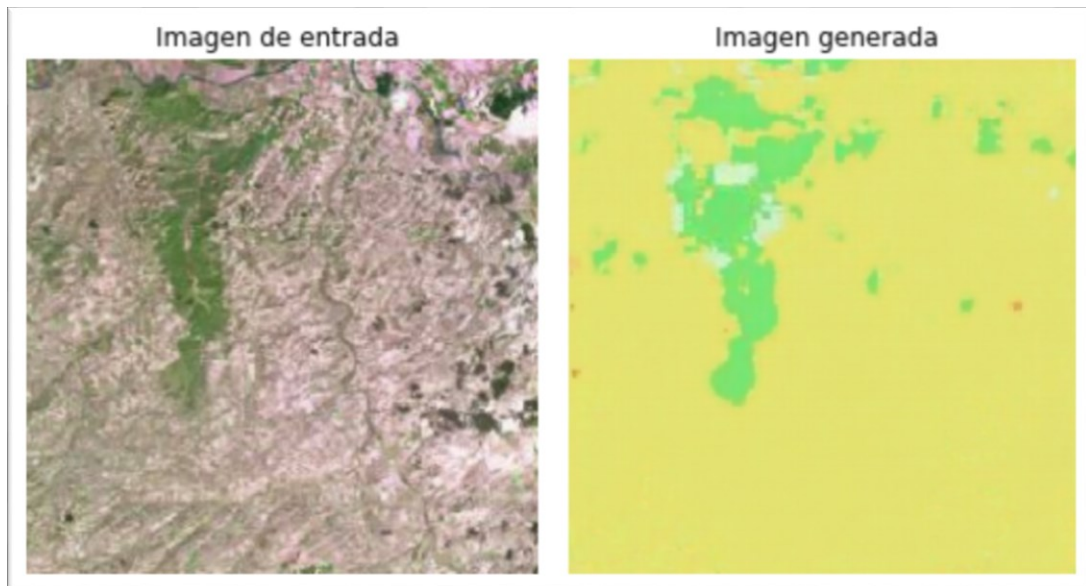


Figura 1.2. Mapa de usos del suelo generado a partir de una ortofotografía del Sentinel 2 mediante el modelo DeepMapsSuperficies
Fuente: elaboración propia, a partir de datos del IGN

Cabe mencionar que en esta investigación se han creado las primeras inteligencias artificiales desarrolladas específicamente para dotar de color y corregir defectos presentes en ortoimágenes en blanco y negro, así como que, a día de hoy, no existe ningún sistema que proporcione resultados comparables con los obtenidos aquí. La comparativa entre otros modelos entrenados para la coloración de imágenes y los modelos desarrollados durante este estudio está recogida en el capítulo 3.



Figura 1.3. Restauración de ortofotografía aérea del vuelo Americano Serie A (1945-1946) mediante la aplicación del modelo DeepsMapsRestaura17
Fuente: elaboración propia, a partir de datos del IGN

Para ello, el procedimiento propuesto se basa en la idea de entrenar nuevos modelos de redes neuronales, restringiendo el aprendizaje al dominio de los datos cartográficos suministrados por el IGN, ofreciendo varias ventajas en comparación con otros métodos tradicionales de tratamiento de imágenes, incluidos aquellos que usan redes neuronales entrenadas con propósitos más generalistas.

Las mejoras obtenidas en los resultados con los modelos entrenados durante la elaboración de este documento son comprobables, empíricamente, con varios experimentos en diversos dominios de aplicación, que se muestran en los entrenamientos de los capítulos del 2 al 6. La aplicación práctica, de las principales inteligencias artificiales desarrolladas en esta investigación, en casos de uso reales, viene recogida en el capítulo 7.

Otras aportaciones al campo de la cartografía son la creación de las primeras inteligencias artificiales capaces de corregir el enfoque de las ortoimágenes aéreas, mejorar su resolución o eliminar el ruido existente en los datos.

Asimismo, se ha creado la primera inteligencia artificial entrenada específicamente para generar los datos faltantes en algunas ortoimágenes (DeepMapsInpainting).

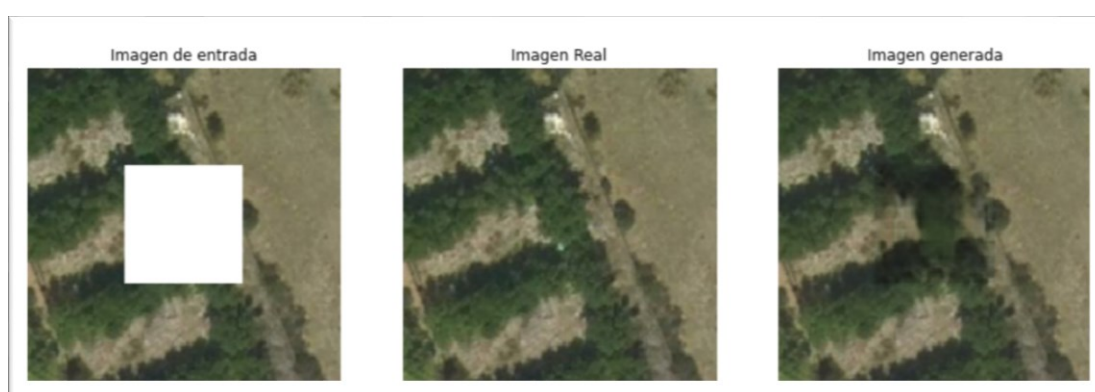


Figura 1.4. Ejemplo de imagen generada (en la que se completa la información de los píxeles ausentes) mediante la aplicación del modelo DeepMapsInpainting, frente a la imagen real
Fuente: elaboración propia, a partir de datos del IGN

El modelo DeepMapsInpainting, supone además una contribución a tener en cuenta en lo referente al tratamiento de ortoimágenes, presentando además potenciales aplicaciones militares. Esta inteligencia artificial es capaz de completar las áreas carentes de datos, debidos a la aplicación de parches de censura, con datos verosímiles, mejorando notablemente el aspecto de la ortoimagen tratada y dificultando enormemente la labor de identificar aquellas áreas sensibles que habían sido originalmente censuradas. Este tipo de aplicación puede comprobarse en el segundo epígrafe del capítulo 7 de este documento, en el que además de las correcciones de las censuras, se recogen algunos ejemplos de ortoimágenes aéreas tratadas con el modelo DeepMapsOculta, también de creación propia; este modelo tiene la capacidad de ocultar edificios y estructuras de creación humana, haciéndolos pasar por

elementos naturales, lo cual tiene claras aplicaciones militares y resuelve la problemática de censurar determinadas áreas de forma mucho más verosímil que mediante las técnicas de parches o difuminados actuales.

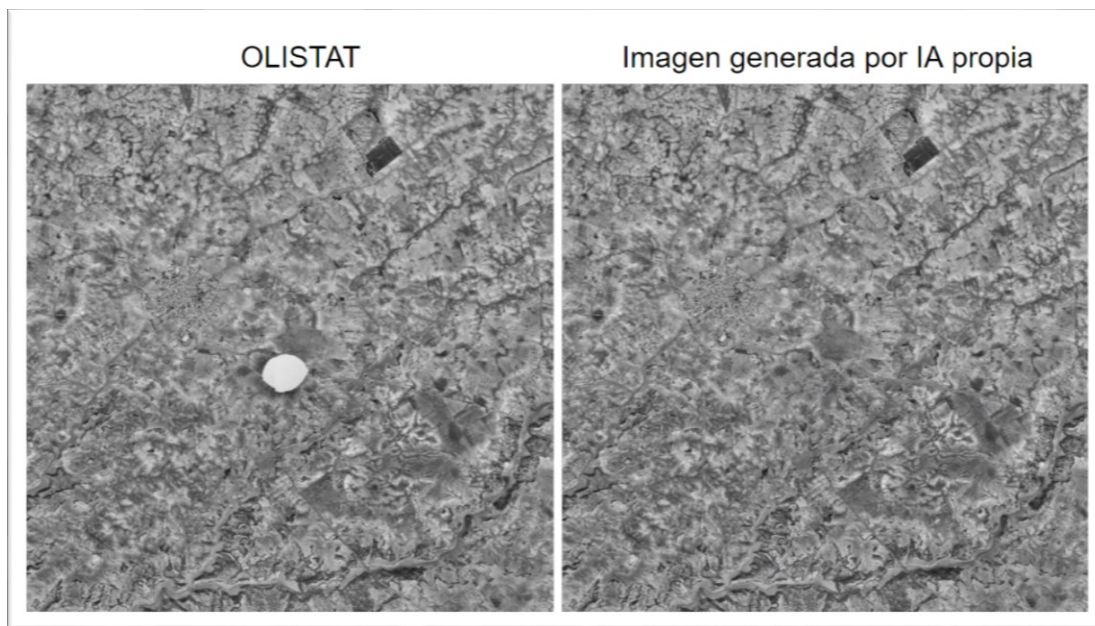


Figura 1.5. Ejemplo de imagen generada (en la que se completa la información de los píxeles ausentes) mediante la aplicación del modelo DeepMapsInpainting, frente a la imagen real con la censura del Escuadrón de Vigilancia Aérea nº2 cercano a Ocaña (Toledo)
Fuente: elaboración propia, a partir de datos del IGN

Debido a los resultados obtenidos en las pruebas realizadas con diferentes arquitecturas de redes neuronales, se ha optado por el entrenamiento de modelos para la creación de inteligencias artificiales ad hoc, las cuales se basan en las arquitecturas que han ofrecido resultados más prometedores en las pruebas realizadas. La creación de estas nuevas inteligencias artificiales viene recogida en los siguientes capítulos.

Por lo tanto, la aportación no consiste únicamente en la creación de inteligencias artificiales capaces de realizar de forma automatizada procesos necesarios en el trabajo cartográfico, sino en haber encontrado una arquitectura que se adapta bien a las necesidades de este tipo de tareas y la elaboración de una metodología innovadora para su aplicación en cartografía.

Se ha comprobado cómo puede extenderse el conocimiento generado por los humanos en un área concreta, como es el caso de la limitación del Copernicus Land Monitoring Service y del CORINE (los cuales están limitados a Europa) a cualquier lugar del planeta mediante el uso de la inteligencia artificial.

Otra contribución, tanto al campo de la cartografía como al de la Inteligencia Artificial, es el descubrimiento de autoemergencia de comportamientos, que en muchos caso resultan útiles, como por ejemplo la generación espontánea de nuevas categorías híbridas y gradientes, para representar mejor la realidad, tanto en los mapas temáticos como en los mapas de uso de superficies y de densidad de árboles o la capacidad para eliminar elementos extraños en las ortoimágenes, aunque los modelos no hubiesen sido entrenados expresamente para ello.

Para comparar el rendimiento de los modelos generados, se han llevado a cabo varias pruebas en ortoimágenes aéreas suministradas por el IGN y bien conocidas por la comunidad cartográfica, como es el caso de las ortofotografías del Vuelo Americano, del cual se muestra un ejemplo a continuación.

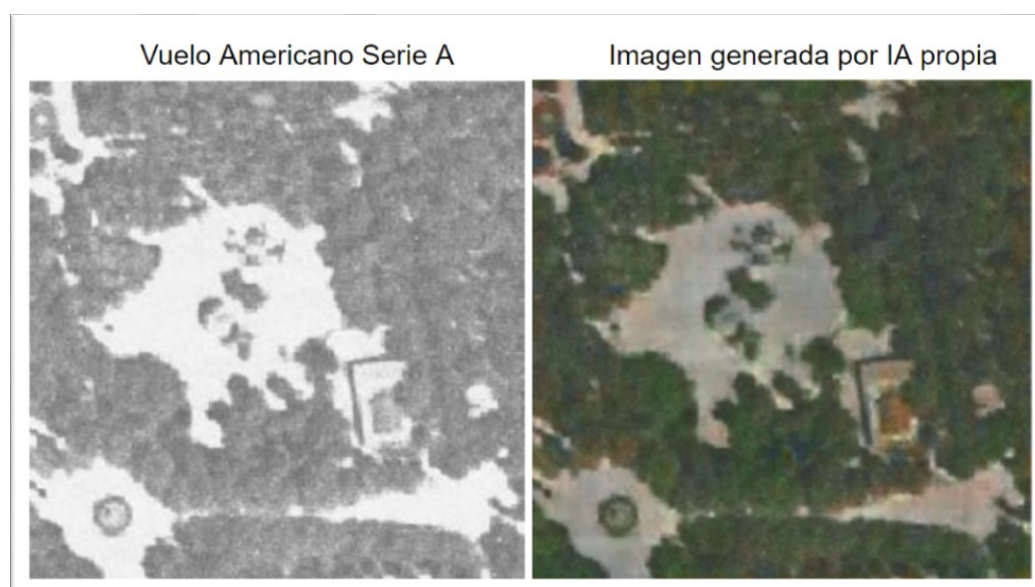


Figura 1.6. Ejemplo de restauración de ortoimagen del Vuelo Americano Serie A (1945-1946) mediante la aplicación de la Inteligencia Artificial DeepMapsRestaura17
Fuente: elaboración propia, a partir de datos del IGN

Además de las inteligencias artificiales anteriormente mencionadas, se han creado otras herramientas, de las cuales destacan dos, una online y otra de escritorio, que permiten la manipulación y el visualizado de grandes volúmenes de datos utilizando técnicas propias del Big Data.

En cuanto a las herramientas elaboradas, en primer lugar, la herramienta online desarrollada es un SIG (<http://cartotecnia.com/teledeteccion/>), a través del cual puede estudiarse el relieve a nivel mundial y que permite, además, la utilización de realidad virtual, para una mejor percepción de éste.

Este software, a su vez, permite la manipulación y superposición de las distintas capas que se han incluido, así como la activación y desactivación de determinados elementos, para adecuarlo a las capacidades de los dispositivos desde los que a él se acceda, teniendo como finalidad, no sólo la utilización de éste para la realización del presente estudio del relieve, sino también para poder ofrecer, de forma pública y gratuita, una herramienta que permita el análisis del relieve. Las instrucciones de uso y el proceso de elaboración de este software vienen detallados en los capítulos 8 y 9, del presente documento.

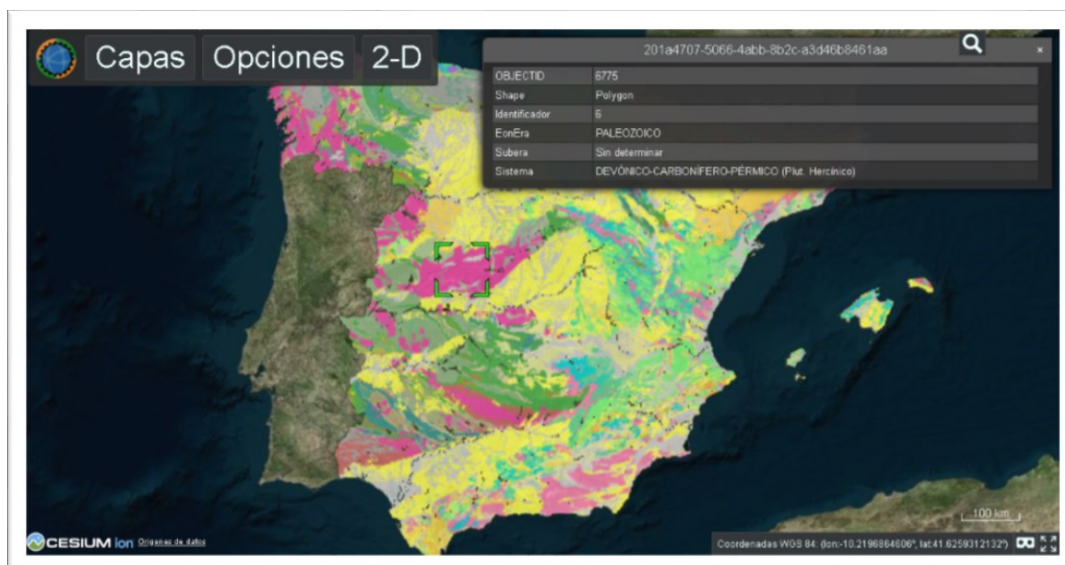


Figura 1.7. Imagen de visor web con capa de edades a escala 1:1.000.000
Fuente: elaboración propia, a partir de datos del IGN y del IGME

En segundo lugar, la herramienta de escritorio ha sido creada para el estudio más detallado de geomorfologías de menor tamaño y para la identificación de estructuras de origen antrópico, aprovechando al máximo la resolución de los datos de las nubes de puntos ofrecidas por el IGN. Esta herramienta, a la que se ha dado el nombre de “Morillas Easy LiDAR” (MEL), ha sido diseñada para realizar el procesamiento masivo de los datos LiDAR, a partir de datos directamente descargados del IGN, con extensión .laz (sin necesidad de la descompresión de éstos en formato .las). Este software permite al usuario llevar a cabo las operaciones que se realizan más habitualmente con nubes de puntos o modelos digitales del terreno, como la manipulación tridimensional de los modelos, el filtrado de los datos, la generación de hillshades, etc. Los detalles de este software, el proceso seguido para su elaboración, así como las indicaciones necesarias para su utilización, son descritos en los capítulos 8 y 9 de este documento.

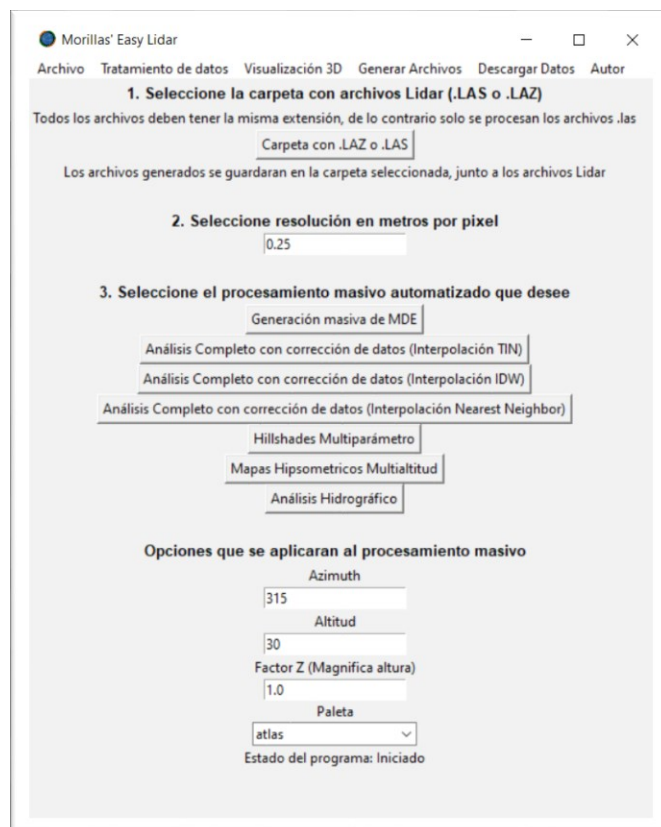


Figura 1.8. Interfaz de la herramienta desarrollada para tratamiento y visualización de datos LiDAR

Fuente: elaboración propia, a partir de datos del IGN

1.5. Estructura de la tesis

La presente tesis se ha constituido en diversas partes, con una estratificación jerárquica, la cual es necesario detallar en este epígrafe. Las secciones de mayor rango son los capítulos, los cuales a su vez están divididos en epígrafes y subepígrafes.

- El capítulo 1 es la Introducción, en el cual se hace un breve avance de los temas a tratar.
- El capítulo 2 se centra en la descripción de las inteligencias artificiales desarrolladas en esta investigación para su aplicación en cartografía. En este capítulo se describe la metodología seguida para el entrenamiento de los modelos propuestos, así como la arquitectura en la que éstos se basan. También incluye algunos ejemplos de los resultados obtenidos con las mismas.
- El capítulo 3 trata la cuestión de añadir color a las ortoimágenes en escala de grises, estudiando la aplicabilidad de las tecnologías existentes y también las aquí desarrolladas para esa labor, siendo, tanto la aplicación de tecnologías existentes como la creación de inteligencias artificiales propias, innovaciones en el campo de la cartografía.
- El capítulo 4 se centra en la mejora de la resolución y el enfoque en el trabajo cartográfico; para ello se analizan los algoritmos existentes, la posibilidad de aplicación de nuevas tecnologías y la eficacia de las inteligencias artificiales desarrolladas durante la investigación en esta tarea.
- El capítulo 5 se ocupa de la corrección automatizada de errores y artefactos mediante la aplicación del Deep Learning, analizando los

resultados obtenidos con modelos existentes cuando se aplican sobre ortoimágenes aéreas, y también los logros obtenidos con los modelos entrenados durante la elaboración de este documento.

- En el capítulo 6 se trata la posibilidad de generación automatizada de mapas de ocupación del suelo mediante las inteligencias artificiales desarrolladas durante esta investigación.
- El capítulo 7 recoge casos prácticos de aplicación de las inteligencias artificiales de creación propia sobre datos diferentes a aquellos con los que fueron entrenadas. Estos casos muestran distintas aplicaciones prácticas que van desde la eliminación de defectos y artefactos o mejora de la calidad de las ortofotografías hasta la generación automática de mapas temáticos, pasando por la restauración de fuentes cartográficas antiguas en escala de grises y bastante deterioradas, a las que se les proporciona color y calidad, haciéndolas similares a las ortofotografías actuales. En este capítulo puede comprobarse la aplicabilidad real de las redes aquí creadas.
- El capítulo 8 trata del proceso de elaboración de las dos herramientas desarrolladas para facilitar el estudio del relieve, tanto la herramienta de escritorio para el tratamiento y manipulación de datos LiDAR como el visor cartográfico online de realidad virtual.
- El capítulo 9 se centra en la metodología de trabajo propuesta para el análisis del relieve mediante el uso de las herramientas aquí desarrolladas para el tratamiento y manipulación de datos tridimensionales.
- El capítulo 10 recoge algunos casos de uso de las herramientas de creación propia, para el tratamiento y visualización de datos

tridimensionales, en el ámbito del análisis del relieve y la identificación de estructuras tanto de origen natural como antrópico.

- Por último, en el capítulo 11 se incluyen las principales conclusiones derivadas de la presente investigación en cuanto a la aplicación de la inteligencia artificial para la automatización de los procesos cartográficos, el entrenamiento de modelos con ortoimágenes y la aplicación de técnicas de Big Data para el procesamiento de datos cartográficos tridimensionales, como las nubes de puntos o los modelos digitales de elevaciones.

Tras las conclusiones se incluyen los anexos, el anexo I muestra la evolución de distintas inteligencias de creación propia, en él pueden verse resultados producidos por estas redes neuronales en fases tempranas de su entrenamiento. El anexo II incluye una muestra de algunos de los archivos generados por el software MEL al ejecutar las opciones de generación de Hillshade Multiparámetro e Hipsométrico Multiparámetro, donde se crean hillshades y mapas hipsométricos con distintas combinaciones de Altitud, Azimuth, Factor Z y paleta de colores (en el caso de los hipsométricos), para de esta forma poder escoger la configuración en la que mejor se aprecian los relieves que se estén estudiando. Y por último el anexo III incluye algunos ejemplos de mapas hipsométricos generados automáticamente mediante la aplicación del software MEL. A continuación de los anexos, se recoge la bibliografía consultada para la realización de esta tesis.

En lo referido a la numeración de las imágenes incluidas en este documento, las distintas figuras están identificadas por dos números; el primero corresponde al capítulo al que pertenecen y el segundo es el número de imagen dentro de ese capítulo; es decir, la Figura 1.8 corresponde a la imagen número 8 dentro del primer capítulo.

1.6. La problemática de las ortoimágenes antiguas

Al trabajar en cartografía con imágenes antiguas nos surgen una serie de problemas, algunos de ellos han sido solucionados hace tiempo, como los diferentes sistemas de referencia de coordenadas, cuya solución se basa en la aplicación de algoritmos matemáticos relativamente simples y que está ya muy extendida en prácticamente la totalidad de los sistemas de información geográfica. Pero, a su vez, existen problemas cuya resolución ni se había planteado hasta el día de hoy debido a la complejidad técnica que éstos entrañan, por ejemplo, poder dar color a ortoimágenes cuyo origen son imágenes en blanco y negro. Esto podría proporcionarnos una visión más realista del terreno en épocas pretéritas y un mejor aprovechamiento de los datos disponibles.



Figura 1.9. Ejemplo de restauración de ortofotografía aérea del Vuelo Americano Serie A (1945-1946) con el modelo DeepMapsRestaura17
Fuente: elaboración propia, a partir de datos del IGN

Hoy en día, gracias al desarrollo de diversas técnicas y a la mejora de la capacidad de procesamiento de nuestras máquinas, podemos plantearnos la resolución de este problema desde distintos enfoques, siendo el principal y el más obvio la aplicación de Deep Learning o aprendizaje profundo, en concreto las redes neuronales convolucionales, las cuales, gracias a su capacidad para analizar imágenes y generar clasificaciones, pueden ser entrenadas para asignar colores realistas a imágenes en blanco y negro. Esto no es algo trivial porque, a pesar de que teóricamente disponemos de tecnologías capaces de realizar dicha tarea, nos enfrentamos a una serie de retos, los cuales se detallarán a lo largo de los siguientes capítulos. Para encontrar la mejor solución al problema se han tenido que realizar distintas pruebas con distintas configuraciones de redes neuronales, así como con distintos entrenamientos de estas, para finalmente encontrar las más adecuadas y corregir algunas de las carencias que surgieron en el procedimiento.

Durante los próximos capítulos se analizaron las distintas posibilidades para llevar a cabo estas tareas, especificando los pros y los contras de cada una de ellas, para finalmente decantarnos por la que mejores resultados nos ofrece.

1.7. La creación automática de mapas de usos del suelo

Las labores de clasificación de los elementos presentes en imágenes se conocen como segmentación semántica. Es en esta clasificación semántica en la que se basan los escasos proyectos existentes de clasificación de ortoimágenes aéreas, ya que, a pesar de ser una tecnología ampliamente explotada en el campo de la Inteligencia Artificial y la visión computarizada en la última década, en la práctica las aplicaciones que se le han dado han sido principalmente de clasificación de imágenes como rostros, personas, objetos, pero salvo alguna excepción no se ha desarrollado este tipo de clasificación automatizada para ortoimágenes aéreas o satelitales; ello se debe a que los modelos existentes en la actualidad han sido entrenados con fotografías cotidianas para los usos más frecuentes que se le están dando a esta tecnología y, debido a la ausencia de ortoimágenes aéreas en los dataset de entrenamiento de dichos modelos, éstos no son eficaces a la hora de aplicarlos a la clasificación de imágenes tomadas desde una perspectiva cenital. De hecho, los proyectos de segmentación de imágenes aéreas mediante Deep Learning eran inexistentes al inicio de esta investigación, siendo únicamente en los tres últimos años cuando han empezado a crearse algunos proyectos, los cuales aún están inmaduros y cuyos resultados son demasiado locales como para una aplicación práctica efectiva.

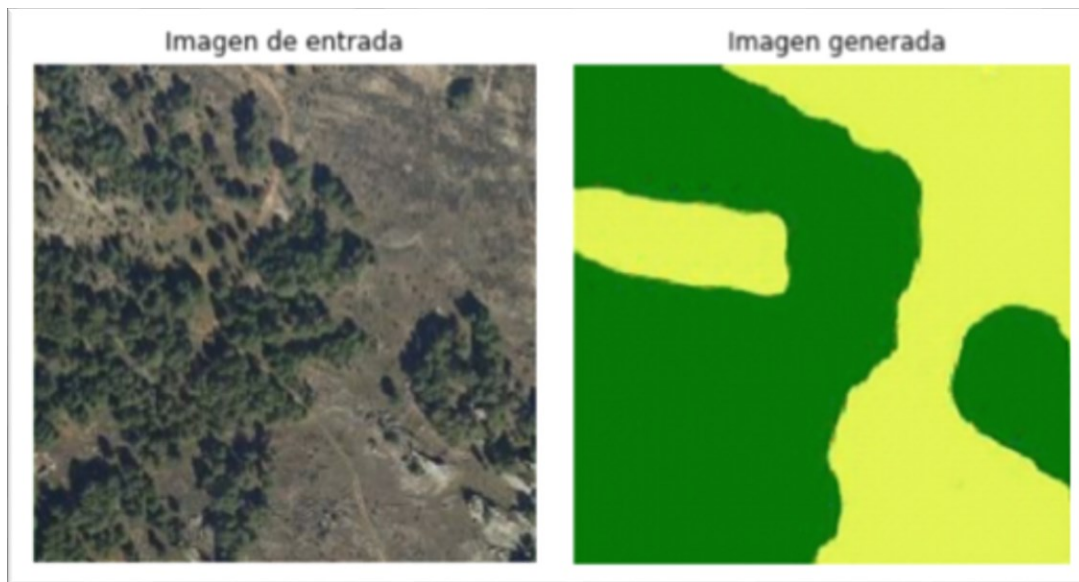


Figura 1.10. Mapa de usos del suelo generado automáticamente por la red neuronal de creación propia DeepMapsUsosNatura
Fuente: elaboración propia, a partir de datos del IGN

Las labores de creación de mapas de usos del suelo hasta ahora se hacían de forma manual, mediante la revisión de ortoimágenes aéreas o satelitales y la aplicación de distintos filtros sobre las mismas. Además, en la mayoría de los casos, al menos en lo que a vegetación se refiere, era necesario el uso de información complementaria a las imágenes registradas en el espectro visible, ya fuese datos procedentes de imágenes multiespectrales o directamente las distintas intensidades con las que se reciben los datos LiDAR. Es por ello por lo que la segmentación automática de imágenes aéreas en el espectro visible posee un gran potencial en teledetección y la generación automatizada de mapas de usos del suelo. Después de la aplicación de los métodos de redes neuronales convolucionales, la precisión de los algoritmos de segmentación semántica podría superar fácilmente el 80%, según afirman P. Isola y su equipo en su trabajo de 2016 “Image-to-image translation with conditional adversarial networks”, siempre que se proporcione un conjunto de datos adecuado para su entrenamiento. A pesar de este éxito, el despliegue de un modelo de segmentación preentrenado, como los existentes hasta la fecha, para estudiar un nuevo terreno que no esté incluido en el conjunto de entrenamiento disminuye significativamente la precisión; esto es debido a que la mayor parte de los

trabajos que se han realizado a este respecto se basan en el entrenamiento de modelos partiendo de imágenes obtenidas localmente mediante el uso de drones, lo cual implica orígenes de datos distintos, con características diferentes. Además, la inmensa mayoría de las redes entrenadas lo han sido con datos urbanos para su uso en aplicaciones de seguridad y no para la generación automatizada de mapas de usos de suelos.

Por lo tanto, la aplicabilidad de los modelos existentes queda restringida a la reducida área de donde se extrajeron los datos de entrenamiento y no sería extensible al territorio español debido al cambio de dominio de las fuentes de datos; es decir, debido a las diferencias entre el conjunto de datos con los que se entrenó el modelo y el nuevo dominio del objetivo derivado de las imágenes de nuestro territorio nacional.

A pesar de que la inmensa mayoría de los modelos han sido entrenados con imágenes urbanas de orígenes diferentes, como ya se ha indicado, existen excepciones en forma de algunos experimentos realizados con imágenes satelitales como las procedentes del LANDSAT. En esta investigación el planteamiento es diferente, ya que se pretende aprovechar la existencia de planes de ortofotografía aérea, como el PNOA, para entrenar la red y por lo tanto poder procesar todos los datos nacionales, debido a la uniformidad existente en estos planes. Sin embargo, el objetivo planteado es aún más ambicioso y, mediante la adaptación de dominios, se plantea la posibilidad de aplicar las redes neuronales entrenadas con datos del PNOA a otras ortoimágenes aéreas nacionales, incluidas las imágenes antiguas en escala de grises. Esto abre la puerta a la creación automatizada de mapas de usos del suelo partiendo de mapas que, en ocasiones, tienen varias décadas. También supone otra innovación el hecho de poder partir de datos comunes recogidos en el espectro visible, ya que, hasta el momento, para detectar las distintas ocupaciones del suelo por parte de especies vegetales, se debía recurrir a ortoimágenes aéreas multiespectrales y utilizar el rango del infrarrojo en lugar del espectro visible, lo que facilitaba la tarea, pero impedía la aplicación de esas técnicas sobre

capturas que no estuvieran en ese rango del espectro. Por supuesto, hoy en día, gracias a la reciente tecnología LiDAR, la labor de identificación de vegetación resulta más sencilla, motivo por el cual en este documento se ha incluido también una herramienta para el procesamiento de las nubes de puntos procedente de esa tecnología (MEL); esta herramienta devuelve unos resultados superiores en resolución a las existentes hasta la fecha. Los detalles de la implementación de la herramienta, así como ejemplos de casos de uso, pueden consultarse en los capítulos 8, 9 y 10.

Antes de avanzar en este capítulo es conveniente ponernos en antecedentes del estado de la técnica, en la cual ya se han hecho algunos progresos en la coloración automatizada de imágenes, la mejora de la resolución o el enfoque, pero los cuales aún no han sido aplicados a la cartografía digital, teniendo esta aplicación una serie de retos y, a la vez, una serie de ventajas que la convierten en un objeto de estudio especialmente interesante.

1.8. La inteligencia artificial

Ya con el surgimiento de las primeras máquinas programables, como las de Lovelace y Babbage en el siglo XIX, la gente comenzó a preguntarse si en algún momento las máquinas podrían llegar a ser inteligentes, lo cual actualmente se está consiguiendo, al menos, en algunos campos de lo que consideramos inteligencia, como el reconocimiento de imágenes, el reconocimiento de voz, análisis predictivos, etc. Hoy en día la Inteligencia Artificial es un campo con innumerables aplicaciones prácticas y de investigación.

En sus inicios la inteligencia artificial pudo abordar y resolver con relativa facilidad problemas que nos resultan difíciles a los humanos, problemas que pueden ser matemáticamente descritos con facilidad y/o que siguen unas reglas sencillas. La dificultad para estas inteligencias artificiales surge en la resolución de tareas que a los humanos nos resultan sencillas pero que son difíciles de describir formal o matemáticamente, ya que son tareas que nosotros realizamos de forma intuitiva, como el reconocimiento facial o del habla, mencionados anteriormente.

En el procesamiento de datos cartográficos existen varios de estos problemas complejos y difíciles de describir matemáticamente. Por ello, con el propósito de dar solución a los mismos, se ha optado por permitir que los ordenadores aprendan de la experiencia y entiendan el mundo en términos de una jerarquía de conceptos, estando cada concepto definido a partir de otros conceptos más simples. Al reunir el conocimiento a partir de la experiencia, es decir, de los datos que se le suministra al ordenador para su entrenamiento, se evita la necesidad de que el programador tenga que definir formalmente todos los conocimientos que va a necesitar el ordenador para realizar la tarea que se le ha asignado. Esta jerarquía de conceptos da la oportunidad al ordenador de aprender conceptos complicados, los cuales son contruidos a partir de otros

más simples. Al construir los conceptos más complejos por encima de los simples es necesaria la creación de muchas capas convolucionales; por ese motivo, a este tipo de aprendizaje en la Inteligencia Artificial se conoce como aprendizaje profundo o Deep Learning.

Algunos de los primeros éxitos de las inteligencias artificiales se dieron en entornos muy formalizados y que no requerían que los ordenadores tuvieran un gran conocimiento del mundo; como ejemplo de ello pueden mencionarse el sistema de juego de ajedrez Deep Blue, desarrollado por IBM y que derrotó al campeón mundial Gary Kasparov en el año 1997. Aquel logro temprano de la Inteligencia Artificial fue posible debido a que el ajedrez es un universo simple, compuesto únicamente por 64 lugares diferentes y 32 piezas que sólo pueden moverse de una forma bien definida. Debido a ello, el juego del ajedrez puede describirse con una lista muy simple de reglas formales, siendo las mismas previamente suministradas por el programador. Esto es una prueba más de la facilidad que poseen los ordenadores para llevar a cabo tareas abstractas, pero bien formalizadas matemáticamente, las cuales normalmente entrañan gran dificultad para los humanos; sólo actualmente la mayor complejidad de las máquinas les ha permitido equipararse a los humanos en tareas cuya descripción en términos matemáticos resulta problemática, como el reconocimiento y procesamiento visual en el cual se centran las investigaciones llevadas a cabo durante la elaboración de este documento.

Debido a que los humanos realizamos estas tareas de reconocimiento visual y procesamiento de imágenes de forma innata e intuitiva, nos resulta difícil definir las formalmente en base a un conjunto de reglas, es ahí donde entran los datos de entrenamiento, mediante los cuales los ordenadores son capaces de recibir ese conocimiento que los humanos tenemos de forma innata sobre el mundo y de esta forma poder comportarse de manera inteligente. El hecho de transmitir ese conocimiento, que no puede ser plasmado de manera formal en un programa a un ordenador, resulta uno de los mayores retos de la Inteligencia Artificial.

Durante años han existido numerosos proyectos de Inteligencia Artificial que han intentado definir esos conocimientos del mundo real e introducirlos mediante reglas formales a través de la codificación, pero ninguno de ellos ha tenido un gran éxito. De las dificultades derivadas de este tipo de desarrollos, actualmente se ha comprendido que los sistemas de inteligencia artificial necesitan la capacidad de adquirir sus propios conocimientos, ellos mismos son capaces de identificar patrones partiendo de los datos en bruto, en lugar de codificar los conocimientos de forma manual; esto es lo que se conoce como Machine Learning.

El Machine Learning es lo que permitió a los sistemas de inteligencia artificial abordar problemas relacionados con el conocimiento del mundo real y poder tomar decisiones que parecen subjetivas. Para que estos sistemas funcionen correctamente, los datos que se le suministran deben ser relevantes para la tarea asignada. La dependencia de las representaciones es un fenómeno que surge en toda la informática e, incluso, en nuestra vida cotidiana. En la informática algunas operaciones, como puedan ser la búsqueda de una colección de datos, pueden realizarse exponencialmente más rápido si la colección está estructurada e indexada de manera inteligente. La gente puede hacer fácilmente aritmética con números arábigos, pero tendrá mayores dificultades si pretende hacerlo con números romanos; por lo tanto, no resulta sorprendente que la elección de las representaciones de la realidad tenga un gran efecto en la eficiencia y el rendimiento de los algoritmos de aprendizaje automático.

Muchas tareas de Inteligencia Artificial pueden ser resueltas diseñando el conjunto correcto de características a extraer para una tarea concreta y luego proporcionando estas características a un sistema de aprendizaje automático. Sin embargo, surge el problema de que para muchas tareas es difícil saber qué características deben ser extraídas; por ejemplo, en el caso de querer desarrollar un programa que sea capaz de identificar los coches en las imágenes, sabemos que los coches poseen ruedas, por lo que es razonable querer utilizar la

presencia de ruedas como característica propia del coche. Desgraciadamente, resulta complicado describir exactamente cómo se ve una rueda en términos de valores de píxeles; a pesar de que una rueda tiene una forma geométrica simple, su imagen puede complicarse por distintos motivos, como las sombras que caen sobre la rueda, la iluminación, el ángulo desde el que fue tomada la imagen, las partes metálicas de la rueda, etc. Una solución a este problema es permitir que la máquina aprenda a descubrir, no únicamente la interrelación que hay entre la representación de la realidad y la salida que proporciona el algoritmo, sino también la representación en sí misma. A este enfoque se le conoce como aprendizaje de la representación, el cual suele proporcionar mejores resultados que los obtenidos con las representaciones diseñadas a mano, permitiendo, además, que los sistemas de inteligencia artificial se adapten rápidamente a nuevas tareas, con un mínimo de intervención humana.

Los algoritmos de aprendizaje de representaciones son capaces de descubrir un buen conjunto de características para una tarea simple en minutos, o para una tarea compleja en horas o incluso meses. Sin embargo, por regla general, el resultado obtenido es mucho más satisfactorio al utilizar estas representaciones aprendidas en lugar de las diseñadas a mano; a esto hay que añadir que el diseño manual de características es una tarea compleja, dado que se requiere una gran cantidad de esfuerzo y de tiempo por parte de los programadores, pudiendo llegar, incluso, a consumir décadas de los desarrolladores.

El mejor ejemplo de un algoritmo de aprendizaje de representación es el autoencoder, el cual combina una función codificadora, que convierte los datos de entrada en una representación diferente y una función decodificadora, que convierte la representación obtenida, de nuevo, en el formato original. Los autoencoders están entrenados para preservar tanta información como sea posible cuando una entrada se ejecuta a través del codificador y luego del decodificador, pero también están entrenados para hacer que la nueva representación tenga varias características que se buscaban.

Debido a lo anteriormente mencionado, las inteligencias artificiales implementadas utilizan un autoencoder, el cual debe, además, desenmarañar los factores de variación y descartar aquellos que no nos interesan, como el ruido, pixelaciones, desenfoques y artefactos en las fotografías, en el caso de las redes neuronales implementadas se ha utilizado para el autoencoder de las inteligencias artificiales de esta tesis la arquitectura U-Net, la cual viene descrita en el capítulo 2.

El proceso de extracción de las características de alto nivel y abstractas, a partir de los datos en bruto, puede llegar a ser muy difícil; muchos de estos factores de variación pueden ser identificados, únicamente, utilizando una comprensión elevada de los datos, casi a nivel humano. Cuando resulta tan difícil obtener una representación como la propia resolución del problema original, a primera vista el aprendizaje de representación no parece resultar de gran ayuda, sin embargo actualmente, gracias a la capacidad de procesamiento disponible para el entrenamiento de las redes neuronales y a la gran cantidad de datos de ortoimágenes que existen, sólo es necesario el diseño de una arquitectura apropiada y la selección adecuada de los datos de entrenamiento para permitir a las inteligencias artificiales aprender por sí mismas una representación correcta de la realidad. Para que esta representación se adapte a nuestros criterios humanos ha sido fundamental la inclusión de un modelo discriminador en la arquitectura de redes desarrolladas.

En la actualidad existen muchas configuraciones y arquitecturas de Deep Learning, pero en esta introducción debe hacerse una mención especial a las GAN (Redes Generativas Antagónicas), ya que la arquitectura utilizada para diseñar los modelos entrenados se corresponde a una variedad de Red Generativa Antagónica Condicional (cGAN), conocida como image-to-image o Pix2Pix.

Las redes generativas antagónicas son una clase de algoritmos de Inteligencia Artificial utilizados en el aprendizaje automático no supervisado que

poseen arquitecturas de redes neuronales profundas, compuestas a su vez por dos redes (generador y discriminador) las cuales se enfrentan una contra la otra. El objetivo principal de las redes generativas antagónicas es generar datos desde cero, como dibujar un retrato, componer una sinfonía o, como en el caso de las redes creadas en esta investigación, generar una ortoimagen aérea o un mapa de usos del suelo. Los modelos generados pertenecen a una variedad de este tipo de redes, conocida como cGAN o Redes Generativas Antagónicas Condicionales, ya que los resultados que arrojan están condicionados por unos datos de entrada, como puede ser una ortofotografía aérea o un mapa de usos del suelo.

Para comprender las redes generativas antagónicas, primero debemos comprender qué es un modelo generativo. En el aprendizaje automático, las dos clases principales de modelos son generativos y discriminativos. Un modelo discriminativo es aquel que discrimina entre dos (o más) clases diferentes de datos, por ejemplo, una red neuronal convolucional que está entrenada para emitir un 1 dada una imagen de un árbol y un 0 en caso contrario. Un modelo generativo, por otro lado, no sabe nada sobre las clases de datos; en cambio, su propósito es generar nuevos datos que se ajusten a la distribución de los datos de entrenamiento.

En función de lo anteriormente expuesto, el funcionamiento de estas redes puede entenderse como un juego en el que el generador intenta producir algunos datos de la distribución de probabilidad y el discriminador actúa como un juez. El discriminador decide si la entrada proviene del verdadero conjunto de datos de entrenamiento o de datos falsos generados. Por otra parte, el generador intenta optimizar los datos para que puedan coincidir con los datos de entrenamiento reales. Podría decirse que el discriminador es una guía del generador para producir datos realistas. Es decir, la idea básica de las Redes generativas antagónicas es configurar un juego entre dos jugadores. Los dos jugadores son generador y discriminador. El generador crea muestras que deben corresponderse con el conjunto de entrenamiento. El discriminador

examina las muestras para determinar si son reales o falsas; es decir, si las muestras de entrada son similares a las del conjunto de entrenamiento o no. El generador está entrenado para engañar al discriminador, es decir, el generador genera una muestra y la pasa al discriminador. El discriminador clasifica la muestra de entrada en dos clases (real o falsa); engañar al discriminador significa que el discriminador clasificará la muestra generada por el generador como real en lugar de falsa. Y de aquí es de donde viene el nombre "antagónicas" en las redes generativas antagónicas, ya que ambas redes se comportan como adversarios.

2. Inteligencias Artificiales Cartográficas

En este capítulo se analizan tanto la arquitectura común a todas las inteligencias artificiales creadas durante esta investigación como la metodología seguida para su entrenamiento y aplicación. Los detalles de los distintos entrenamientos vienen recogidos en los capítulos siguientes.

El mayor aporte que supone esta tesis es el conocimiento sobre la aplicación de Inteligencia Artificial y Big Data a la cartografía y el establecimiento de algunas bases para la adaptación de estas tecnologías a los datos cartográficos. Pero, además, todos los modelos entrenados están planteados para que tengan una utilidad práctica, de forma tanto independiente como conjunta, ya que abordan problemas habituales en la cartografía. La mayoría de ellos han sido entrenados para que, al resolver estos problemas, realicen también una adaptación de dominio que permita la aplicación de los modelos entrenados para la generación automatizada de mapas de usos del suelo a ortoimágenes aéreas de orígenes distintos a los que se utilizaron para el entrenamiento.



Figura 2.1. Ejemplo de generación de un mapa de usos del suelo aplicando una de las Inteligencias Artificiales de creación propia (DeepMapsSuperficies)
Fuente: elaboración propia, a partir de datos del IGN

Esto resulta muy útil, ya que al aplicar un algoritmo de Machine Learning generalmente se asume que los datos de entrenamiento y los datos de la prueba

pertenecen a la misma distribución subyacente. Pero en los escenarios reales, nos enfrentamos a ciertas discordancias entre ellos. Esta discordancia disminuye la eficacia del modelo cuando se enfrenta a datos que pertenecen a otro dominio. La adaptación del dominio es un campo separado en el aprendizaje automático, que tiene como objetivo rectificar esta discordancia y ayudar a que el modelo pueda generalizarse mejor y ser efectivo al aplicarse a datos de diferentes dominios.

Estudios recientes se orientan principalmente a mejorar la adaptabilidad de los algoritmos de Deep Learning. Muchos trabajos en este campo utilizan datos simulados; de hecho, éstos buscan utilizar la adaptación del dominio para mejorar la eficiencia en la segmentación de las imágenes reales mediante el entrenamiento de modelos con datos sintéticos, estrategia que también se ha seguido en los desarrollos incluidos en esta investigación.

Debido a que existen distintos desafíos y procesos que se pretenden abarcar en este documento, en base a la revisión bibliográfica y a las pruebas realizadas, de las cuales las más relevantes se encuentran recogidas en los capítulos posteriores, se ha optado por recurrir a una arquitectura que fuese lo suficientemente versátil como para poder resolver todos los retos que se plantean en este estudio y, a la vez, tuviese una efectividad comprobada. Es por ello por lo que se ha optado por las redes generativas antagónicas, concretamente las redes generativas antagónicas condicionales conocidas como image-2-image o, más coloquialmente, como Pix2Pix.

A diferencia de otras redes profundas, basadas en modelos preentrenados de visión computarizada como ResNet, AlexNet, VGG16, VGG19, etc., Pix2Pix no requiere definir ninguna relación entre los dos tipos de imágenes. No hace suposiciones sobre la relación y, en su lugar, aprende el objetivo durante el entrenamiento, comparando las entradas y salidas definidas durante el entrenamiento e infiriendo el objetivo. Esto hace que Pix2Pix sea muy adaptable

a una amplia variedad de situaciones, incluidas aquellas en las que no es fácil definir la tarea que queremos modelar.



Figura 2.2. Corrección de error debido a la falta de datos mediante la aplicación de la inteligencia artificial DeepMapsInpainting
Fuente: elaboración propia, a partir de datos del IGN

Gracias a la versatilidad de esta arquitectura en cuanto a conversión de imagen a imagen, puede adaptarse la resolución de las distintas labores cartográficas descritas anteriormente en este trabajo simplemente variando el entrenamiento de los diferentes modelos y los datos con los que son alimentados. La recolección y generación de los datos necesarios para el entrenamiento de cada una de las inteligencias artificiales aquí elaboradas supone un reto en sí mismo, estando detallados los procesos seguidos en los distintos epígrafes de este capítulo.

Debido a que la arquitectura utilizada para todas las inteligencias artificiales desarrolladas durante la presente investigación es la misma (solamente se han realizado pequeños ajustes, como el cambio del hiperparámetro LAMBDA), el siguiente epígrafe recoge los detalles de ésta, mientras que en los próximos se especifican las peculiaridades de los distintos entrenamientos llevados a cabo para el desarrollo de las distintas redes creadas, así como los resultados obtenidos con las mismas. El orden en el que se recogen los pormenores de los distintos entrenamientos viene definido por la labor que desempeñan las distintas inteligencias artificiales.

2.1. Entrenamiento de los modelos

Aunque el proceso de entrenamiento de los distintos modelos ha sido diferente en función de la tarea para la que han sido desarrolladas las distintas inteligencias artificiales, todos los procesos comparten la mejora progresiva en los resultados que se iban obteniendo a medida que aumentaba el número de iteraciones.

Una de las conclusiones que se ha podido obtener de los distintos entrenamientos es que, si bien los modelos van mejorando a medida que son entrenados, si el entrenamiento se realiza con una misma muestra de imágenes llega un punto en el que, o bien los resultados empiezan a empeorar o bien la red se vuelve demasiado específica, teniendo dificultades al aplicarla en imágenes distintas de la muestra de entrenamiento; cuanto más se sobreentrene una red más específica se vuelve. Esto también depende, en gran medida, del número de imágenes utilizadas durante el entrenamiento y la variedad de estas.

Cuando el número de ortoimágenes con las que es entrenado un modelo es limitado, o la variedad de estas es escasa, si se entrena durante demasiadas iteraciones se observa que los resultados que devuelve son deficientes al tratar las ortoimágenes similares a aquellas que tenían escasa o ninguna representación en la muestra de entrenamiento. Es decir, que si se entrena una red con 1000 ortoimágenes en las cuales aparecen bosques y únicamente un par de ellas en las que aparecen edificios, a la hora de procesar imágenes en las que aparecen edificios probablemente cometan errores. Por ello es fundamental elegir correctamente una buena representación de todos los elementos que pueden surgir en las tareas que se le vayan a asignar a la red. La ventaja fundamental de entrenar las redes con un número alto de ortoimágenes es que van a ser más versátiles y será menos probable que se encuentren con elementos que no sepan representar correctamente.

En los siguientes capítulos se recogen más detalles de los distintos entrenamientos, correspondientes a las inteligencias artificiales que han devuelto mejores resultados. Debido a que se trata de redes profundas dedicadas a labores cartográficas, se las ha bautizado como DeepMaps, y dado que se han entrenado diferentes modelos para la realización de tareas diversas, el nombre de cada una de ellas incluye su función y características, en un intento de que sea, en la medida de lo posible, autodescriptivo. Un ejemplo de lo mencionado anteriormente sería la inteligencia artificial DeepMapsColor19, la cual tiene como objetivo la coloración de ortoimágenes aéreas y ha sido entrenada con imágenes obtenidas con un nivel de zoom de 19, o lo que es lo mismo una escala aproximada de 1:1.000. Así mismo, se ha podido comprobar que las redes entrenadas con ortoimágenes con una escala mayor devuelven mejores resultados, ya que con una resolución mayor los elementos individuales como árboles, edificios, vehículos, etc., son más fácilmente identificables tanto por los humanos como por las inteligencias artificiales aquí creadas, sin embargo tienen el inconveniente de tener que procesar un gran número imágenes para tratar un área reducida, por lo que es necesario encontrar el equilibrio entre estos dos factores, en función del área que se vaya a tratar.

2.2. Metodología seguida para el entrenamiento de los modelos propios

Debido a que no existían redes generativas antagónicas condicionales específicamente entrenadas para la realización de labores cartográficas a partir de ortoimágenes, ha sido necesario crearlas, para lo cual se ha recurrido a recopilar los datos de entrenamiento mediante la generación de teselas de 256x256 píxeles, y el tratamiento de las mismas mediante scripts de Python, creados, asimismo, expresamente para adaptarlas a las distintas tareas en las que se iban a entrenar las redes y así poder crear los pares de imágenes necesarios para el entrenamiento de este tipo de redes. A continuación, se muestra un ejemplo de un par de imágenes generado para el entrenamiento.



Figura 2.3. Par de imágenes generado para el entrenamiento de uno modelo de coloración creados

Fuente: elaboración propia, a partir de datos del IGN

En primer lugar, debían seleccionarse los datos de los que se iba a partir; en nuestro caso, los datos proceden de distintas fuentes para las distintas redes entrenadas, pero la mayoría de ellas parten de los datos del PNOA de Máxima Actualidad proporcionados por el IGN. La metodología aquí planteada es la que

se ha utilizado para entrenar todos los modelos desarrollados en este estudio, existiendo pequeñas variaciones entre unos y otros, pero salvo variaciones en el número de imágenes utilizadas, número de iteraciones y el origen de los datos en algunas, todas pueden ajustarse al esquema planteado.

La primera tarea es adquirir los datos; para ello se ha recurrido al centro de descargas del IGN, desde el cual se ha procedido a la selección y descarga de los datos.



Figura 2.4. Sección de descarga de datos del PNOA de Máxima Actualidad del IGN
Fuente: elaboración propia, a partir de datos del IGN

Una vez seleccionados los datos de los que vamos a partir, se descargan los archivos correspondientes a la zona con la que se van a entrenar. En el ejemplo mostrado se han descargado los datos correspondientes a las hojas MTN50, correspondientes a la Comunidad de Madrid.

División administrativa | Hoja MTN50 | Coordenadas | R. Catastral

Comunidad Autónoma ▼

MADRID, COMUNIDAD DE ▼

Buscar

Figura 2.5. Panel de selección de datos por división administrativa
Fuente: elaboración propia, a partir de datos del IGN

Para agilizar el proceso de descarga se recurre a la descarga masiva, seleccionando todos los archivos con el formato que queremos descargar, para lo cual, tras realizar la búsqueda, se filtran los resultados para que sólo nos muestre los archivos en formato .ecw.

Filtro sobre los resultados

Tipo fichero: **ECW** (Todos)

Buscar en resultados: Escriba texto para buscar en los resultados **Buscar**

Nombre	Formato	Tamaño (MB)	Fecha	Localizar	Descargar
PNDA-MA-OF-ETRS89-HU30-H50-0458.ECW	ECW	1924.09	08/2017, 09/2017, 09/2018		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0483.ECW	ECW	2592.61	08/2017, 09/2018		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0484.ECW	ECW	1917.40	08/2017, 09/2017		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0485.ECW	ECW	2199.32	08/2017, 09/2017, 09/2018		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0507.ECW	ECW	2255.72	08/2017, 09/2018		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0508.ECW	ECW	1677.60	08/2017, 09/2017		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0509.ECW	ECW	2207.43	08/2017, 09/2017		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0510.ECW	ECW	2002.94	08/2017, 09/2018		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0532.ECW	ECW	1964.90	08/2017, 09/2018, 10/2018		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0533.ECW	ECW	2035.73	08/2017		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0534.ECW	ECW	2206.27	08/2017		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0535.ECW	ECW	1873.17	08/2017		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0557.ECW	ECW	1747.44	08/2017, 09/2018, 10/2018		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0558.ECW	ECW	2133.54	08/2017		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0559.ECW	ECW	2048.10	08/2017		Añadir
PNDA-MA-OF-ETRS89-HU30-H50-0559.ECW	ECW	2122.14	08/2017		Añadir

Figura 2.6. Filtrado de los datos para proceder únicamente a la descarga de ficheros ECW
Fuente: elaboración propia, a partir de datos del IGN

Tras esto, se seleccionan todos los archivos de las páginas mostradas como resultado, haciendo clic en el símbolo +, que se aprecia en la esquina superior derecha de figura 2.6.

Una vez seleccionados todos los archivos que van a ser utilizados, se accede a la cesta de descargas y se inicia la descarga mediante un pequeño programa en Java.



Figura 2.7. Página de descarga de los datos seleccionados
Fuente: elaboración propia, a partir de datos del IGN

Seguidamente se ejecuta el programa descargado, habiendo antes indicado la localización donde se quieren guardar los archivos y se inicia la descarga.

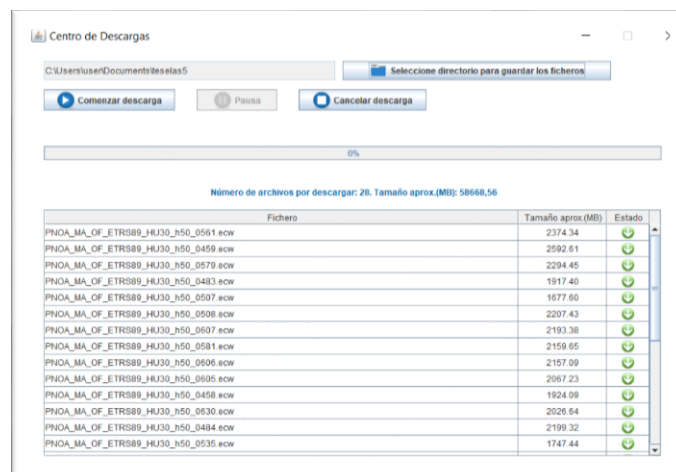


Figura 2.8. Interfaz del programa Java para la descarga de los datos seleccionados
Fuente: elaboración propia, a partir de datos del IGN

Una vez descargados los archivos, éstos se cargan en el software de tratamiento de datos geográfico QGIS, en el cual se crea un nuevo proyecto y se cargan las distintas capas descargadas, teniendo en cuenta establecer el SRC tanto del proyecto como de las distintas capas en el sistema de referencia identificado con el número 25830 (ETRS89 / UTM zona 30N - Proyectado), ya que los datos vienen en el sistema de referencia geodésico global ETRS89 en el huso 30, según el sistema de coordenadas universal transversal de Mercator o UTM que divide la Tierra en husos de 6° de longitud cada uno, generados a partir del meridiano de Greenwich y numerados en orden ascendente hacia el este. El huso 30 en este sistema de proyección abarca el centro de la Península Ibérica y por lo tanto la Comunidad de Madrid. El hecho de que los datos descargados desde el IGN vengan en ETRS89, es decir en el Sistema de Referencia Terrestre Europeo 1989, se debe a que desde el 27 de agosto de 2007 España adoptó este sistema de referencia geodésico global, sustituyendo al anterior sistema de referencia geodésico regional empleado en Europa, el ED50 (European Datum 1950).

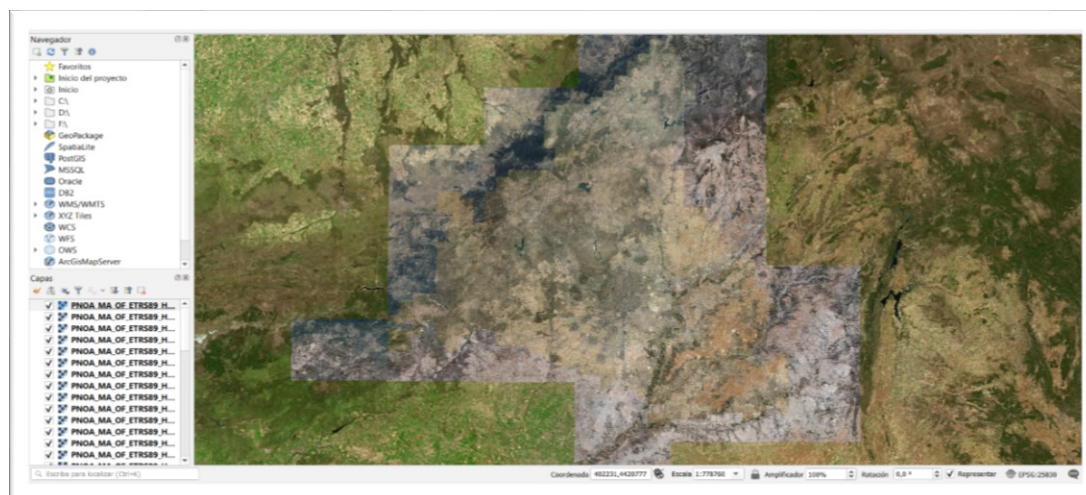


Figura 2.9. Archivos del PNOA de Máxima Actualidad descargados del IGN y cargados en QGIS, correspondientes a la Comunidad de Madrid
Fuente: elaboración propia, a partir de datos del IGN

Para generar las distintas teselas a partir de una de las capas descargadas se recurre al proceso de GDAL, para tratamiento de datos ráster, GdalAteselas

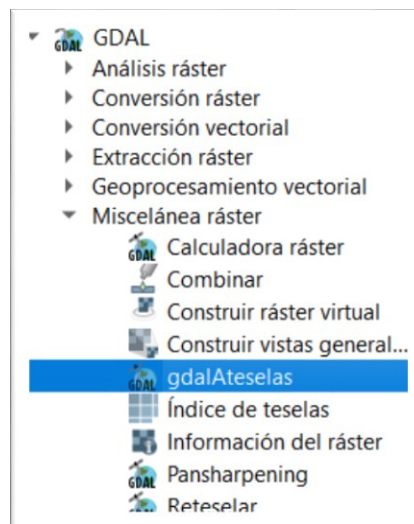


Figura 2.10. Selección del proceso de transformación de datos ráster a teselas
Fuente: elaboración propia con QGIS

Antes de comenzar el proceso, debe seleccionarse la capa a partir de la cual se van a generar las teselas y los niveles de zoom que se van a representar; en el caso de este ejemplo, el nivel de zoom con el que se generarán los datos de entrenamiento es 19, es decir con una escala aproximada de 1:1.000.

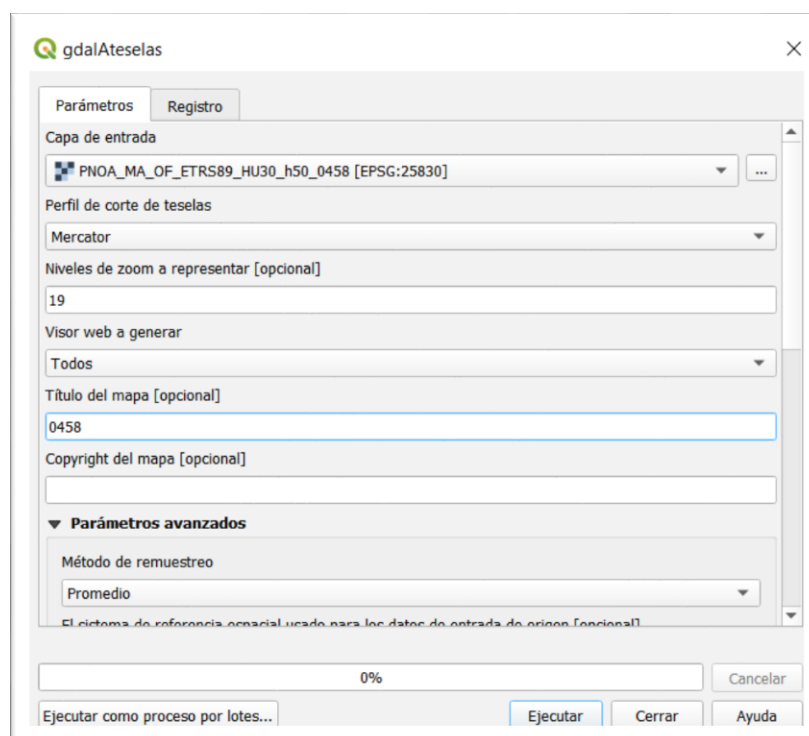


Figura 2.11. Generación de teselas mediante QGIS
Fuente: elaboración propia con QGIS

Una vez ejecutado el proceso tenemos los archivos de las teselas ordenados por carpetas.

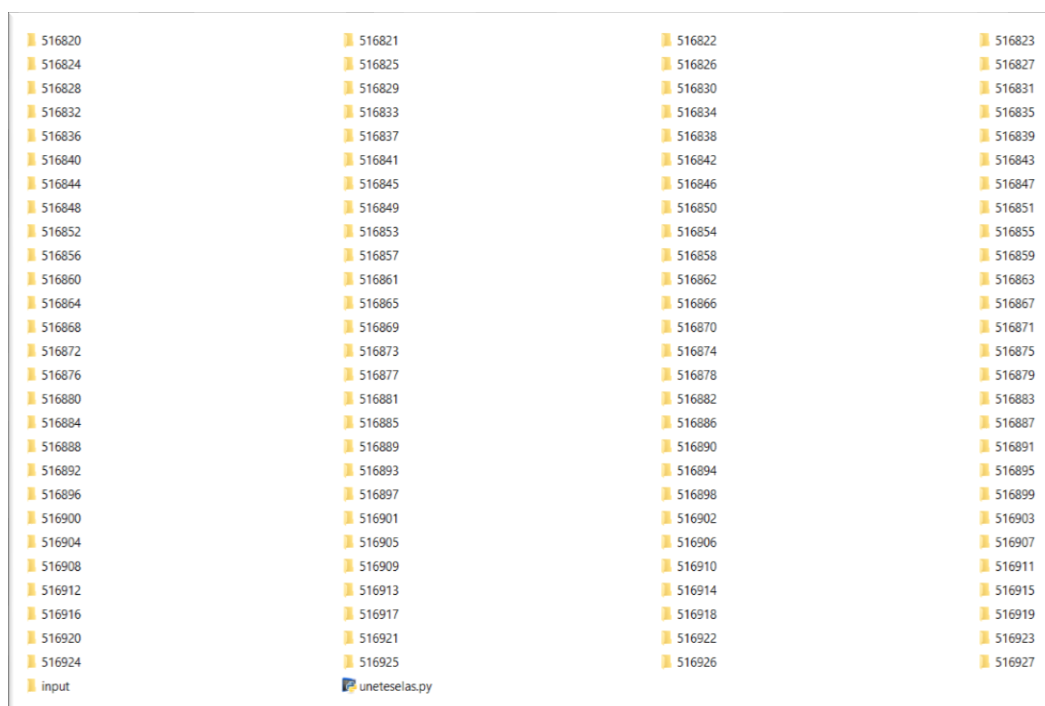


Figura 2.12. Muestra de parte de las carpetas con teselas generadas, script para la unión de estas y carpeta input que contiene todas las teselas renombradas y reunidas en una única carpeta.

Fuente: elaboración propia.

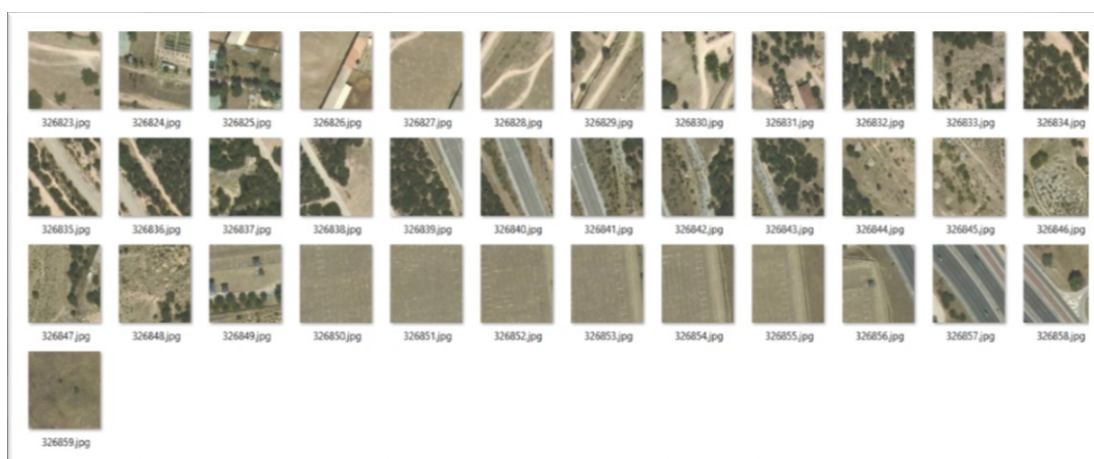


Figura 2.13. Ejemplo de teselas generadas
Fuente: elaboración propia, a partir de datos del IGN

Como las teselas vienen separadas en carpetas y los nombres de archivo se repiten, se crea un pequeño script que asigna un nombre diferente a cada archivo de imagen en función de su nombre y ruta, y los copia en una carpeta que se crea, si no existe, llamada “Input”.

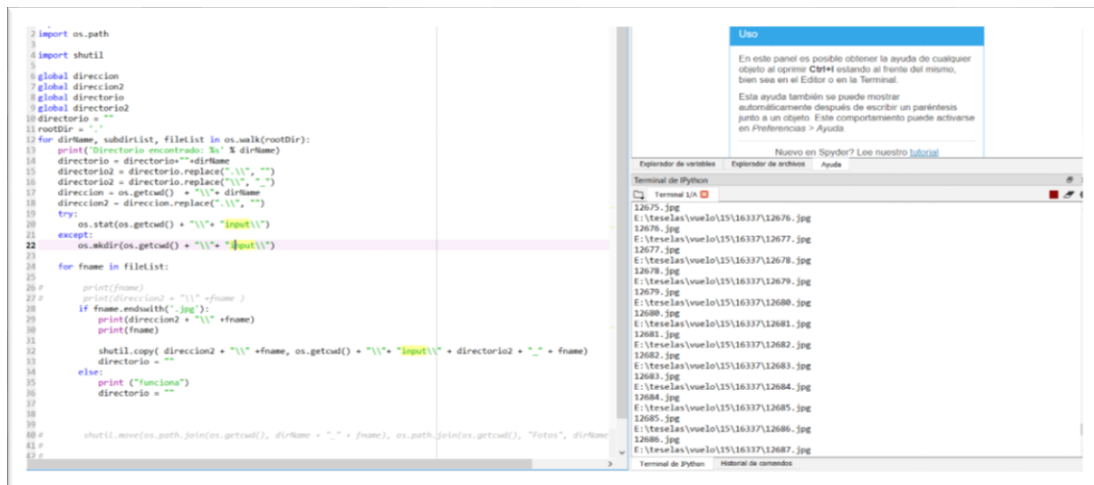


Figura 2.14. Ejecución del script de Python para renombrar los archivos de imagen y reunirlos en una única carpeta

Fuente: código de elaboración propia, entorno de desarrollo Spyder

Esta carpeta contiene todas las teselas generadas de 256x256 píxeles, que son las que se van a utilizar para entrenar los distintos modelos.

La necesidad de recurrir a la utilización de teselas se deriva de las limitaciones computacionales actuales; los algoritmos utilizados, tanto en el entrenamiento de los modelos como en la aplicación de éstos, consumen más recursos computacionales que los algoritmos más simples para el tratamiento de imágenes mediante la aplicación de filtros. Además, a este problema se une la necesidad, tanto para el entrenamiento como para la aplicación de los modelos, de que exista un contexto, ya que las inteligencias artificiales aquí desarrolladas no se limitan a realizar operaciones sobre píxeles individuales o sobre vecindades reducidas de píxeles, sino que analizan las imágenes en su conjunto para, de esta forma, al igual que ocurre en la visión biológica, poder identificar elementos y patrones sobre los que se van a realizar unas u otras

operaciones. Debido a esto, los píxeles no pueden ser analizados de forma secuencial y deben cargarse en memoria imágenes completas, lógicamente a mayor tamaño de imagen mayor es la memoria consumida, por lo que el tamaño de éstas debe reducirse lo suficiente como para que puedan ser procesadas.

Hay que tener en cuenta, también, que para realizar un entrenamiento eficaz el número de imágenes que se procese debe ser elevado. Todo esto hace necesario recurrir a teselas, ya que las imágenes completas de las cartografías son inabarcables con la tecnología existente en la actualidad; la elección de las dimensiones de estas teselas responde más a un convencionalismo, puesto que las teselas utilizadas habitualmente en cartografía digital, principalmente en la cartografía web, tienen una extensión de 256x256 píxeles.

Las máquinas utilizadas, tanto para el entrenamiento como para la aplicación de los distintos modelos, tienen capacidad para manejar teselas de mayores dimensiones, las cuales al ofrecer a la inteligencia artificial un contexto más amplio reportan resultados algo más precisos; sin embargo, el tiempo de entrenamiento necesario también aumenta, siendo la medida de 256x256 píxeles la más adecuada de las que se han probado por la relación entre tiempo consumido y calidad de los resultados.

El total de teselas generado en este proceso es demasiado elevado como para hacer factible el entrenamiento en el tiempo disponible para ello, por lo que se ha restringido el número de teselas aplicadas a la mayor parte de entrenamientos, escogiendo estas de forma aleatoria y descartando de esa selección aquellas que no fuesen adecuadas. En base a las pruebas realizadas, se ha podido comprobar que tanto el tiempo de entrenamiento necesario como los resultados obtenidos pueden mejorarse llevando a cabo una selección previa de los datos de entrenamiento.

Para obtener los mejores resultados posibles es importante incluir en los datos de entrenamiento una muestra significativa de todos los elementos que se pretende enseñar a la inteligencia artificial; también es importante que los datos de entrenamiento se acerquen lo máximo posible a los datos que va a tener que procesar la red. Es por ello por lo que los resultados al aplicar la red a datos de fuentes diferentes a la fuente de los datos con los que fue entrenada sufren una gran mejoría si se incluye en la metodología de entrenamiento un preprocesamiento de las imágenes para generar variaciones de estas que se asemejen más a los datos a los que va a tener que enfrentarse en un futuro. Un ejemplo de lo anteriormente expuesto puede apreciarse en la red DeepMapsRestaura17, la cual ha sido desarrollada durante este trabajo y en cuyo entrenamiento se han incluido teselas procesadas mediante un script de Python que genera 128 versiones de cada una de las teselas utilizadas en el entrenamiento; cada una de estas versiones presentaba una combinación diferente de luminosidad, contraste y desenfoque, para que de esta forma la inteligencia artificial pudiese aprender cómo comportarse al tratar con datos de ortoimágenes antiguas en las que, debido al soporte físico de las imágenes, los procesos de captura, revelado, etc., se aprecian irregularidades en estos parámetros.



Figura 2.15. Ejemplo ortofotografía aérea del Vuelo Americano Serie B (1956-1957) del Parque de Roma (barrio de Estrella, Madrid), restaurada con el modelo DeepMapsRestaura17
Fuente: elaboración propia, a partir de datos del IGN

A continuación, se muestran algunos de los ejemplos de los pares de imágenes utilizados para el entrenamiento de DeepMapsRestaura17, en los cuales se han modificado la luminosidad, el contraste y el desenfoque, para que la inteligencia artificial pueda aprender cómo tiene que corregir y dar color a imágenes deterioradas en blanco y negro.

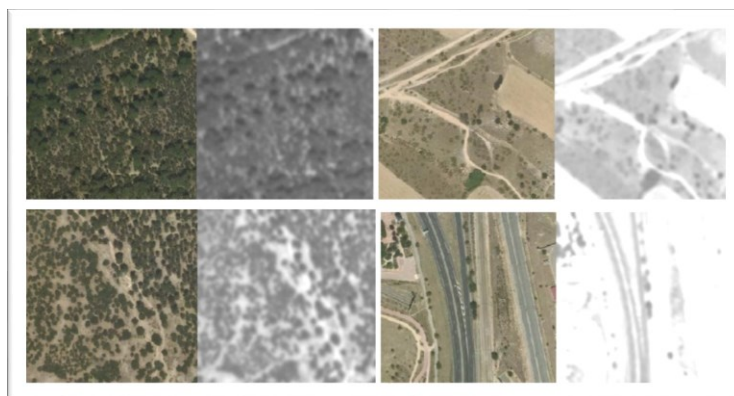


Figura 2.16. Ejemplo de pares de imágenes generados a partir de variaciones de las imágenes originales, utilizados en el entrenamiento de DeepMapsRestaura17
Fuente: elaboración propia, a partir de datos del IGN

Debe añadirse que tras las pruebas realizadas se ha podido comprobar que la generación de variantes de las imágenes mediante la modificación de la luminosidad y el brillo permite que una red neuronal sea entrenada a partir de un número menor de imágenes originales, ya que con las variaciones el número de imágenes en el entrenamiento se multiplica; además permite que los modelos entrenados de esta manera sean más robustos y versátiles. Este fenómeno también se ha podido comprobar con la inclusión de ruido; sin embargo, la generación de variantes de las imágenes mediante brillo y contraste supone una auténtica novedad y un posible avance en los métodos de entrenamiento de las inteligencias artificiales, no sólo en cuanto a su aplicación cartográfica, ya que hasta la fecha no se había utilizado este recurso para optimizar el entrenamiento de redes convolucionales de ninguna clase. Esto podría considerarse de gran utilidad, ya que elimina la necesidad de realizar un pretratamiento a las imágenes que van a ser procesadas por una inteligencia artificial, al ser la propia inteligencia artificial la que internamente consigue entender las imágenes,

independientemente de las variaciones de luminosidad o contraste que puedan existir en los datos de entrada.



Figura 2.17. Ejemplo de teselas de 256x256 píxeles utilizadas para generar los pares de imágenes con los que se han entrenado los modelos
Fuente: elaboración propia, a partir de datos del IGN

Hecho esto, se procede a generar los pares de imágenes con los que se van a entrenar las distintas redes. Esto se hace de forma masiva, utilizando los scripts de Python creados a tal efecto.

```
3 import numpy as np
4 import cv2
5 from matplotlib import pyplot as plt
6 import os
7 import PIL
8 directory = r'./input2/'
9 for filename in os.listdir(directory):
10     if filename.endswith(".jpg") or filename.endswith(".png"):
11         print(os.path.join(directory, filename))
12         foto= cv2.imread (os.path.join(directory, filename))
13         cv2.rectangle(foto,(80,80),(175,175),(255,255,255),-1)
14         cv2.imwrite(os.path.join(directory, filename.replace("png", "jpg")), foto)
15
16
17     else:
18         continue
```

Figura 2.18. Fragmento de código Python utilizado para la creación de las imágenes de entrenamiento del modelo DeepMapsInpainting
Fuente: código de elaboración propia, entorno de desarrollo Spyder

El ejemplo mostrado en la figura 2.18 es el script creado para dibujar un rectángulo blanco dentro de cada tesela y, de esta forma, generar los archivos necesarios para entrenar el modelo de corrección de artefactos debidos a huecos o ausencia de datos.

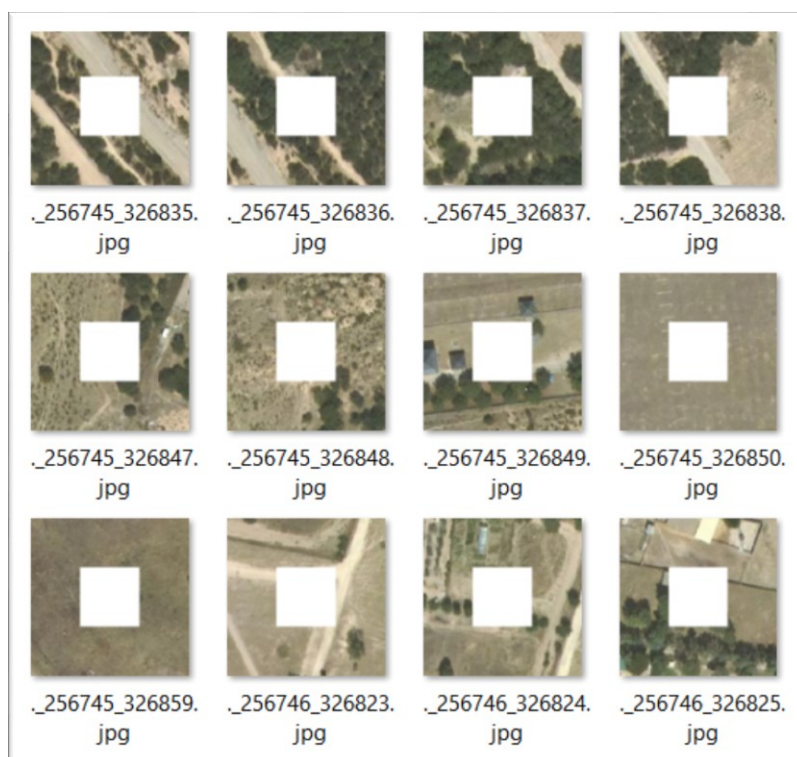


Figura 2.19. Imágenes procesadas utilizadas para el entrenamiento del modelo DeepMapsInpainting

Fuente: elaboración propia, a partir de datos del IGN

El siguiente paso es unir la tesela original con la tesela modificada correspondiente, lo cual se hace mediante un script de Python.

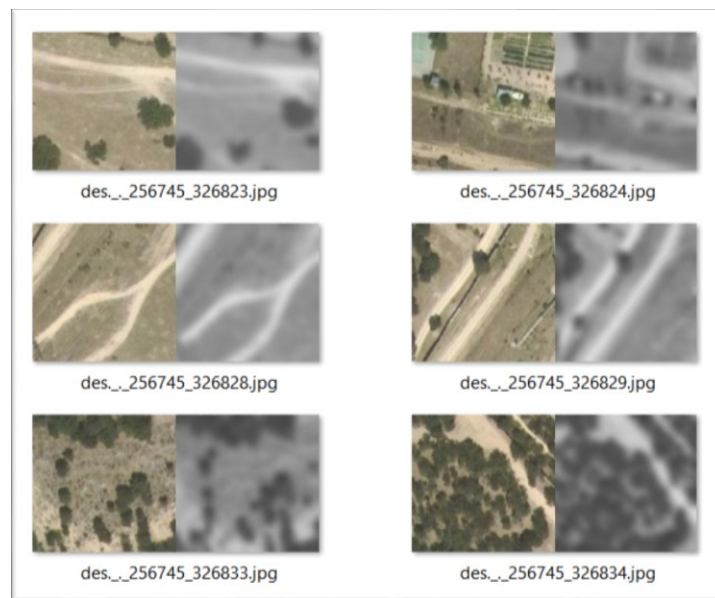


Figura 2.20. Ejemplo de pares de imágenes utilizados para el entrenamiento del modelo DeepMapsOptimiza

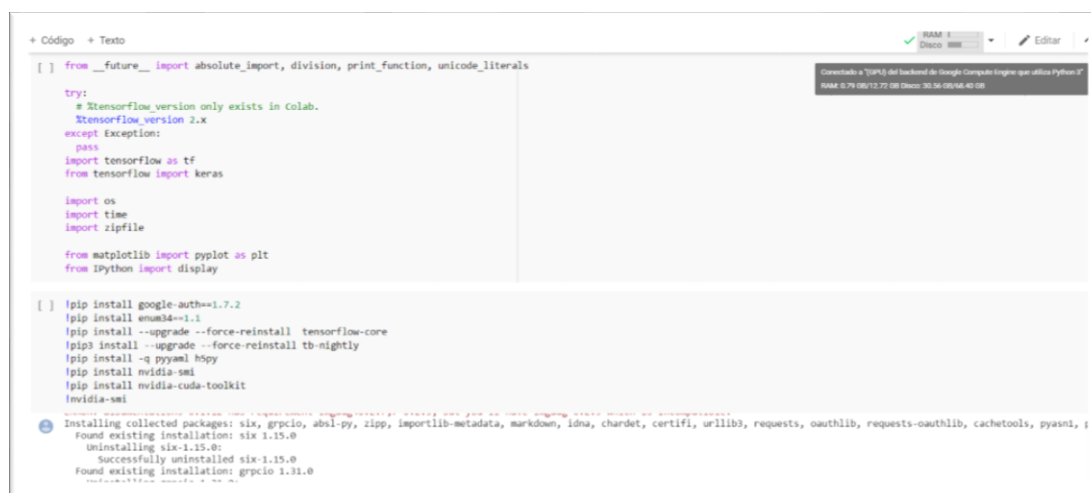
Fuente: elaboración propia, a partir de datos del IGN

En el ejemplo de la imagen superior se muestran los pares de imágenes generados para el entrenamiento de un modelo en las tareas de enfoque y coloración. Lógicamente, para cada tarea se han tenido que generar pares de entrenamiento distintos. Las particularidades de cada uno de ellos son comentadas en los capítulos posteriores.

A continuación, se procede al entrenamiento de los distintos modelos, pero debido a que los equipos disponibles no tienen la potencia de cálculo necesaria, se ha recurrido a la ejecución del código en los servidores de Google Colaboratory, los cuales ofrecen gratuitamente una GPU. El inconveniente está en que no es un servicio estable, por lo que para poder entrenar satisfactoriamente la red, debemos hacer backups de los checkpoints del entrenamiento y alojarlos en un almacenamiento permanente; para ello se ha desarrollado un script que permite que, con cada iteración del entrenamiento, se grabe el modelo y los pesos generados, así como los checkpoints, en Google Drive, desde donde se cargan en la siguiente ejecución del código, de tal forma que el entrenamiento puede así proseguir donde se quedó.

Una vez incluidas todas las imágenes necesarias para el entrenamiento de la red neuronal en una sola carpeta, la cual podemos subir, o bien a Google Drive para que el código consulte las imágenes desde ahí (lo cual resulta lento, ya que debe realizar una conexión a Google Drive por cada imagen que quiera consultar), o podemos comprimir todas las imágenes en un archivo zip, el cual cargamos en el servidor de Google Colaboratory utilizado para el entrenamiento, para después descomprimirlo y acceder a sus imágenes desde la misma máquina que está ejecutando el código. Por su eficiencia, esta última ha sido la opción elegida.

Hecho esto, se ejecuta el código en Python para entrenar la red desde uno de los servidores de Google Colaboratory.



```
+ Código + Texto
[ ] from __future__ import absolute_import, division, print_function, unicode_literals

try:
    # TensorFlow_version only exists in Colab.
    TensorFlow_version 2.x
except Exception:
    pass
import tensorflow as tf
from tensorflow import keras

import os
import time
import zipfile

from matplotlib import pyplot as plt
from IPython import display

[ ] !pip install google-auth==1.7.2
!pip install umsgid==1.1
!pip install --upgrade --force-reinstall tensorflow-core
!pip3 install --upgrade --force-reinstall tb-nightly
!pip install -q pyyaml b2py
!pip install nvidia-smi
!pip install nvidia-cuda-toolkit
!nvidia-smi

Installing collected packages: six, grpcio, absl-py, zipp, importlib-metadata, markdown, idna, chardet, certifi, urllib3, requests, oauthlib, cachetools, pyasn1, s
Found existing installation: six 1.15.0
Uninstalling six-1.15.0:
Successfully uninstalled six-1.15.0
Found existing installation: grpcio 1.31.0
```

Figura 2.21. Carga de las librerías en el script de Python para entrenar los modelos en Google Colaboratory

Fuente: código de elaboración propia, entorno de desarrollo Google Colaboratory

Una vez se ha entrenado durante las suficientes iteraciones, se comprueban los resultados del modelo mediante la ejecución de otro script de Python en el propio servidor de Google Colaboratory.



Figura 2.22. Generación de muestras a partir de los datos de entrenamiento procesadas con el modelo entrenado DeepMapsInpainting.
Fuente: elaboración propia, a partir de datos del IGN.

Es conveniente revisar los resultados en las distintas etapas del entrenamiento para controlar el aprendizaje de las inteligencias artificiales.

A continuación, se muestra un ejemplo de la función de generación de imágenes que permite ir comprobando los resultados producidos por el modelo que se está entrenando y la llamada a la misma.



Figura 2.23. Script para la generación de muestras aplicando el modelo entrenado (en este caso DeepMapsRestarua17) al dataset de pruebas basado en los datos de entrenamiento.

Fuente: elaboración propia, a partir de datos del IGN

En el anexo I se han incluido imágenes generadas en distintas fases de los procesos de entrenamiento de los modelos DeepMapsColor19, DeepMapsInpainting, DeepMapsBosques y DeepMapsBordes17. Este anexo constituye un ejemplo de cómo han ido evolucionando las inteligencias artificiales a medida que han avanzado en su entrenamiento.

Todas las inteligencias implementadas en esta tesis partieron de cero, por lo que antes de comenzar su entrenamiento devolvían datos incoherentes, como la imagen mostrada en la figura 2.24.



Figura 2.24. Ejemplo de ortoimagen generada por el modelo DeepMapsColor19 antes de comenzar su entrenamiento.

Fuente: elaboración propia a partir de datos del IGN.

Las inteligencias artificiales entrenadas para asignar color a las ortofotografías antiguas, en las primeras etapas de su desarrollo comienzan a ser capaces de identificar las luces y sombras de la imagen de entrada, componiendo una representación cada vez más nítida de la realidad a medida que son entrenadas.



Figura 2.25. Ejemplo de ortoimagen generada por la inteligencia artificial DeepMapsColor15 en las primeras fases de su desarrollo.

Fuente: elaboración propia, a partir de datos del IGN.

Una vez que logran representar la ortofotografía de entrada de forma reconocible, empiezan a aprender cómo asignar los distintos colores. Los primeros colores que son capaces de agregar a la imagen de entrada en escala de grises dependen de los datos utilizados en su entrenamiento; en el caso de las inteligencias artificiales aquí creadas, como han sido entrenadas con datos de España, los primeros elementos que aprenden a colorear son la vegetación y la tierra. Al principio, aunque hayan aprendido a colorear los elementos más

comunes en los datos de entrenamiento, siguen cometiendo errores de coloración, tal y como muestra el siguiente ejemplo.



Figura 2.26. Ejemplo de ortofotografía generada por la inteligencia artificial DeepMapsColor19 cuando aún no había adquirido experiencia en la coloración.

Fuente: elaboración propia, a partir de datos del IGN.

Poco a poco, a medida que aumentan las iteraciones del entrenamiento, las inteligencias artificiales empiezan a adquirir otros conceptos, como el de carreteras y tejados, pero tiene dificultades con los elementos menos frecuentes como piscinas o vehículos.



Figura 2.27. Ejemplo de ortofotografía generada por la inteligencia artificial DeepMapsColor19 cuando aún no había alcanzado suficiente madurez.

Fuente: elaboración propia, a partir de datos del IGN.

Finalmente, tras haber entrenado con millones de imágenes, ya que el entrenamiento consta de miles de imágenes y los modelos desarrollados, a su vez, han sido entrenados durante miles de iteraciones, las inteligencias artificiales acaban siendo capaces de identificar los elementos menos

frecuentes, como los coches y, como se muestra en la Figura 2.28, de añadirles color.

A pesar de que estas inteligencias artificiales en inicio eran una “tabula rasa”, en sólo unas cuantas iteraciones, en las que revisaban algunos miles de imágenes, aprendían la tarea para la que estaban siendo entrenadas, volviéndose más expertas y aportando mejores resultados a medida que el entrenamiento avanzaba, superando cualquiera de los sistemas o procedimientos existentes orientados a los mismos fines, sobre todo en cuanto a su aplicación cartográfica se refiere, llegando incluso a superar la calidad de los datos originales con los que se entrenaron, como se comprueba en el siguiente ejemplo, en el cual el modelo generado ha sido capaz de aumentar la calidad de la imagen original tomada como modelo.

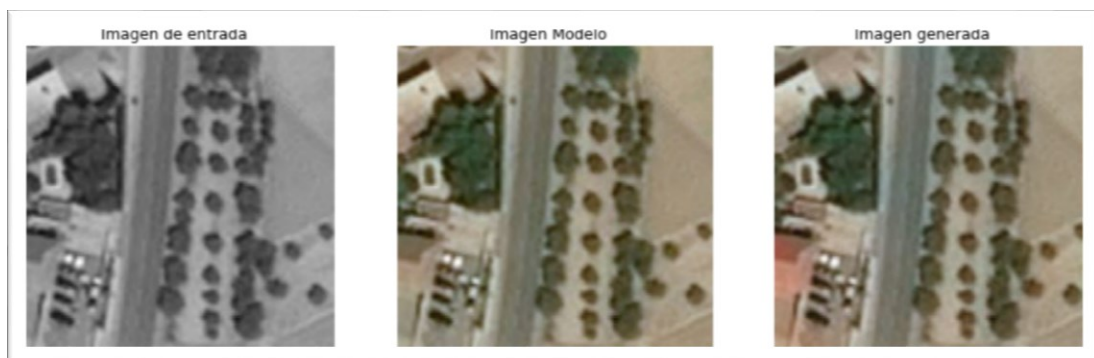


Figura 2.28. Ejemplo de imagen generada por el modelo DeepMapsColor19 al final de su entrenamiento.

Fuente: elaboración propia a partir de datos del IGN.

Otra ventaja observada, a consecuencia de los resultados producidos por las inteligencias artificiales desarrolladas durante este estudio, es que al haber sido entrenadas con imágenes que poseen una coloración correcta, es capaz de mejorar incluso la coloración de ortoimágenes del PNOA en las que, por las condiciones de iluminación en las que fue tomada la imagen o por deficiencias en la captura producidas por cualquier otro motivo, el color no fue correctamente capturado, tal y como se muestra en el ejemplo de la Figura 2.28 en la que la ortoimagen generada por la inteligencia artificial DeepMapsColor19 ha mejorado

el color del tejado que se observa a la izquierda, superando la calidad de la ortoimagen original.

De esta forma, el modelo puede aplicarse, no sólo a dar color a imágenes en blanco y negro, sino también a la corrección de deficiencias en las imágenes de la misma fuente utilizada para su entrenamiento. Lo mejor de esto es que si la calidad de la gran mayoría de las imágenes con las que es entrenado un modelo es buena, aunque algunas tengan deficiencias, la red es capaz de aprender a colorear correctamente, e incluso de corregir, las deficiencias existentes en la muestra de entrenamiento. Este hecho permite la automatización de procesos de mejora de cualquier colección de ortoimágenes sin intervención humana, siempre que la mayoría de los datos de los que se parta para el entrenamiento del modelo tengan un mínimo de calidad.

2.3. La arquitectura de los modelos

Antes de profundizar en la arquitectura de las redes neuronales entrenadas en esta investigación, es conveniente aclarar algunos conceptos que serán necesarios para comprender las descripciones expuestas en este capítulo. En primer lugar, dado que las redes están compuestas por múltiples capas convolucionales pueden ser categorizadas como redes de DeepLearning o de aprendizaje profundo. El hecho de escoger una arquitectura basada en capas convolucionales se debe a que el propósito de las inteligencias artificiales creadas para esta investigación es el procesamiento visual de ortoimágenes. A grandes rasgos puede decirse que las distintas capas de los diferentes modelos que componen las redes neuronales, son conjuntos de lo que en Machine Learning se conoce como neuronas, las cuales tienen unas similitudes importantes con las neuronas biológicas; de hecho su desarrollo fue posible gracias al conocimiento que reportaron los avances en neurofisiología y psicología cognitiva producidos durante la primera mitad del siglo XX, los cuales permitieron la creación de la primera neurona artificial en los años 50 (el perceptrón), lo que permitió la creación de las primeras redes neuronales.

Las neuronas artificiales de las redes neuronales pueden definirse como funciones que imitan el comportamiento de las neuronas biológicas. Uno de los conceptos fundamentales para poder entender la arquitectura de las redes empleadas, es el de función de activación. En la arquitectura de los distintos modelos que componen las redes neuronales utilizadas se han aplicado diferentes funciones de activación (ReLU, LeakyRelu, Tanh y Sigmoide). Todas las funciones de activación utilizadas tienen en común que son una forma de emular el potencial de activación de las neuronas biológicas. Al igual que en biología las neuronas necesitan que el voltaje que reciben llegue a un valor que las activa, en las redes convolucionales la función de activación activa las neuronas de las capas de activación no lineal cuando reciben un valor determinado. Tanto en el caso biológico como en el caso de las redes

neuronales artificiales, una vez activada la neurona, ésta transmite un impulso a las otras neuronas a las que está conectada. Las funciones de activación más empleadas en las redes neuronales utilizadas durante esa investigación son las funciones ReLU ($f(x) = \max(0, x)$) y LeakyReLU ($f(x) = x$ si $x \leq 0$ y αx si $x > 0$ donde α es una constante, en el caso de las redes creadas en esta investigación α tiene un valor de 0,2), debido a que estas funciones de activación son más simples y consumen menos recursos en sus calculos, haciendo que el proceso de entrenamiento sea más rapido que empleando otras funciones de activación más complejas como la Sigmoide ($f(x)=1/(1+\exp(-x))$) o la tangente hiperbólica también llamada Tanh ($f(x)=\tanh(x)$), las cuales aunque también han sido empleadas en la arquitectura utilizada, sin embargo su uso ha sido restringido a lo imprescindible.

Dado que las inteligencias artificiales que se han creado están orientadas al procesamiento de imágenes y, por lo tanto, se basan en redes convolucionales, en las cuales cada capa de convolución debe capturar alguna característica de la imagen y transmitirla a la siguiente capa de convolución. Los valores negativos no son importantes en el procesamiento de imágenes y, por lo tanto, se establecen en 0, sin embargo, los valores positivos deben pasar a la siguiente capa.

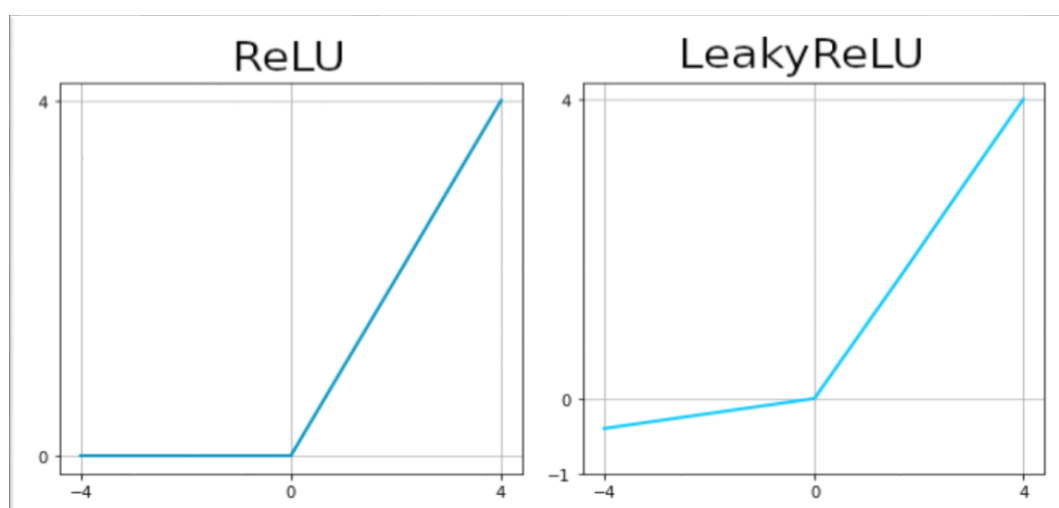


Figura 2.29. Representación gráfica de las funciones ReLU y LeakyReLU (con pendiente debida al factor α de 0,2)
Fuente: elaboración propia

Las distintas capas de los modelos que componen las redes neuronales están conectadas entre sí. Cada una de las capas convolucionales procesa la salida de neuronas que están conectadas en regiones locales de entrada, o lo que es lo mismo, en píxeles cercanos, calculando el producto escalar entre los valores de sus píxeles (estos valores también se conocen como pesos) y una pequeña región a la que están conectados en la capa de entrada. De esta manera, la información viaja de una capa a otra hasta componer un mapa de características, este proceso formará parte del modelo codificador, dentro de los modelos generadores empleados. Los modelos generadores de las redes que se han utilizado, además de tener un modelo codificador, tienen un modelo decodificador, que reconstruye una imagen a partir del mapa de características, en un procedimiento de transmisión de la información entre distintas capas convolucionales similar al de codificación, pero a la inversa. Para mejorar los resultados de las redes generativas tradicionales, en las redes neuronales desarrolladas en esta investigación se ha utilizado para el entrenamiento, además de un modelo generador, un modelo discriminador.

Como ya se ha mencionado al comienzo de este capítulo, la arquitectura empleada para crear las inteligencias artificiales desarrolladas en esta tesis se basa en la arquitectura Image-to-Image, más conocida como Pix2Pix, la cual fue presentada en 2016 por Isola y su equipo en la Universidad de Berkeley en su publicación “Image-to-Image Translation with Conditional Adversarial Networks”.

A esta arquitectura se le han realizado pequeñas modificaciones en función de los propósitos buscados en la creación de las distintas inteligencias artificiales. Las modificaciones realizadas se basan en el ajuste del hiperparámetro LAMBDA, que ajusta la relación entre la pérdida del generador y la pérdida L1, que representa el error absoluto medio entre la imagen generada y la imagen real utilizada como referencia para el cálculo del valor de pérdida total del generador, reduciéndolo en algunos casos del valor de 100 propuesto en la publicación anteriormente mencionada. Para los casos en los que debe hacerse una transformación que no mantiene estrictamente los valores de los

píxeles de la imagen de entrada, como sería el caso de los modelos desarrollados para mejorar la resolución de una imagen muy pixelada o la creación de mapas de usos del suelo, se ha podido comprobar que valores de LAMBDA más bajos ofrecen mejores resultados. En particular, un hiperparámetro LAMBDA de 10 o de 5 mejora en gran medida el realismo de los resultados obtenidos sin perjudicar su fidelidad a los datos de entrada en los supuestos anteriormente mencionados. Lógicamente LAMBDA al ser un hiperparámetro se configura antes del entrenamiento y no forma parte del entrenamiento como tal, es decir es un parámetro que no se obtiene a partir de los datos del entrenamiento, no se utiliza para modelar los datos directamente, pero influye en la capacidad y características de aprendizaje del modelo.

La arquitectura de Red Generativa Antagónica de Pix2Pix implica un modelo generador, un modelo discriminador y un procedimiento de optimización del modelo modulado con los gradientes derivados de las pérdidas del discriminador y del generador. Estos tres componentes trabajan conjuntamente durante cada una de las iteraciones del proceso de entrenamiento, retroalimentándose los unos con los resultados de los otros.

Esta arquitectura utiliza una Red Generativa Antagónica Condicional para aprender una función de mapeo de una imagen de entrada a una imagen de salida. La red está compuesta por dos piezas principales, el generador y el discriminador. El generador transforma la imagen de entrada para obtener la imagen de salida. El discriminador mide la similitud de la imagen de entrada con una imagen desconocida (ya sea una imagen objetivo del conjunto de datos o una imagen de salida del generador) e intenta adivinar si ésta fue producida por el generador. El discriminador y el generador aprenden simultáneamente, y una vez que el generador está entrenado sabe lo suficiente sobre la distribución de las muestras de entrenamiento para que ahora pueda generar nuevas muestras que compartan propiedades muy similares.

Hay que aclarar que todo el código desarrollado se basa en la librería TensorFlow, tanto para entrenar las inteligencias artificiales como para aplicarlas, ya que los tensores que componen las distintas capas de las redes neuronales están creados mediante el uso de las funciones de TensorFlow. Dado que un tensor es un array multidimensional, basado en Numpy (a esta librería no se la llama expresamente en el código desarrollado ya que sus funciones vienen implementadas dentro de la propia TensorFlow). Esta librería cuenta con matrices N-dimensionales. Los tensores son funciones multilineales que cuentan con varias variables vectoriales, mientras que la matriz multidimensional es una estructura de datos que permite representar un tensor en un sistema de coordenadas. Por lo tanto, los tensores son funciones o contenedores que definimos y se encargan de llevar a cabo los cálculos reales, para lo cual hay que alimentarlos con datos, pudiendo representarse los mismos mediante una matriz multidimensional. Un tensor escalar o de 0 dimensiones contiene un valor único y no posee ejes, un tensor de 1 dimensión también se conoce como vector y puede expresarse como una lista de valores, un tensor vectorial o vector tiene 1 eje, y por último un tensor de 2 dimensiones tiene 2 ejes y también se conoce como matriz. El número de dimensiones que tiene un tensor se denomina rango. Los tensores que actúan como funciones se ajustan mediante la asignación de pesos que tiene lugar durante el entrenamiento en base al ajuste de las funciones de pérdida, a medida que las redes procesan nuevos datos. La estructura y dimensiones de las matrices multidimensionales que componen las redes convolucionales, tanto del modelo generador como del discriminador, vienen recogidas a continuación en los siguientes subepígrafes. Es importante aclarar que las funciones de pérdida o loss functions son la clave del entrenamiento de los modelos de redes neuronales. El valor de la función de pérdida es un valor positivo que decrece a medida que la inteligencia artificial está siendo entrenada y el rendimiento del modelo va mejorando. La evolución de los valores de las funciones de pérdida L1 y de pérdida total producidos durante las primeras fases de uno de los entrenamientos viene recogida en los gráficos de las figuras 2.57 y 2.58.

2.3.1. Generador

El generador tiene la labor de tomar una imagen de entrada y realizar la transformación para producir la imagen objetivo. La arquitectura del generador es una red del tipo U-Net modificada. Se utiliza una arquitectura de modelo U-Net para el generador, en lugar del modelo común de Codificador-Decodificador, también conocido como Encoder-Decoder.

El generador, tanto en el caso del modelo Codificador-Decodificador, como en el caso del modelo U-Net utilizado en las redes creadas para esta investigación, puede dividirse en dos subunidades (codificador y decodificador). El codificador toma la imagen de entrada y reduce su dimensión para codificar las características principales en un vector, mientras el decodificador toma este vector y mapea las características almacenadas en una imagen.

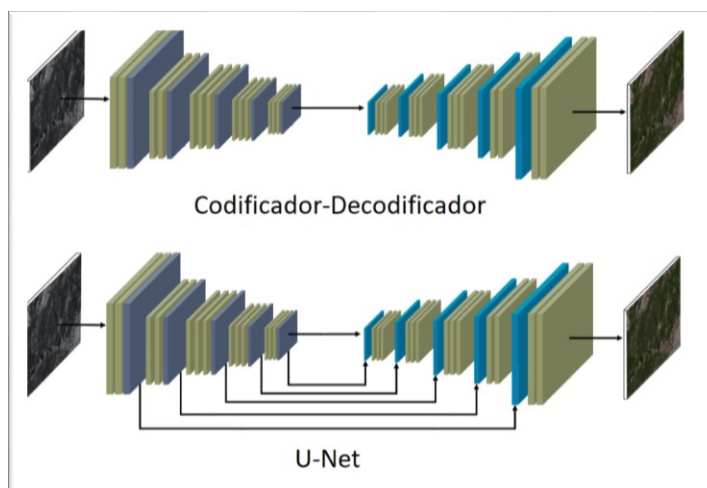


Figura 2.30. Esquema comparativo de las arquitecturas Codificador-Decodificador y U-Net, donde se muestra la conexión entre capas intermedias de la red U-Net
Fuente: elaboración propia

La arquitectura del generador de Codificador-Decodificador implica tomar una imagen como entrada y reducir su muestreo en unas pocas capas hasta una capa de cuello de botella, donde la representación es entonces aumentada de nuevo en unas pocas capas antes de emitir la imagen final con el tamaño deseado.

La arquitectura del modelo U-Net es muy similar en el sentido de que implica reducir la muestra hasta un cuello de botella y aumentarla de nuevo hasta una imagen de salida, pero se realizan enlaces o conexiones de salto entre capas del mismo tamaño en el codificador y el decodificador, lo que permite eludir el cuello de botella.

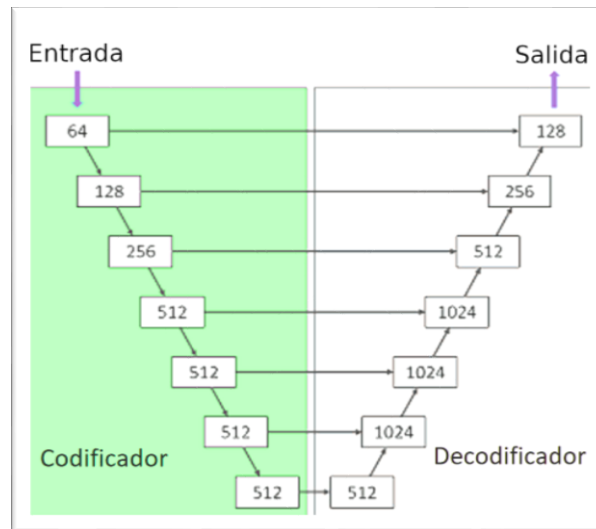


Figura 2.31. Esquema del detalle de la arquitectura del modelo U-Net utilizado
Fuente: elaboración propia

Se puede definir la arquitectura de este modelo U-Net usando la notación abreviada como:

- Codificador: C64-C128-C256-C512-C512-C512-C512
- Decodificador: CD512-CD1024-CD1024-C1024-C1024-C512-C256-C128

La C está referida a cada bloque estándar de Convolución, Normalización por lotes y ReLU (rectified linear unit o unidad lineal rectificada).

La última capa del codificador es la capa de cuello de botella, que no utiliza la normalización por lotes y utiliza una activación ReLU en lugar de LeakyRelu.

Tanto el modelo generador como el discriminador utilizan bloques de capas estándar de Convolución - Normalización por lotes - ReLU, como es común en las redes neuronales convolucionales profundas.

La arquitectura de cada bloque del codificador del modelo generador es la siguiente:

- Convolución --> Normalización por lotes --> Leaky ReLU.

Mientras que la arquitectura de cada bloque del decodificador del modelo generador es la siguiente:

- Convolución transpuesta o convolución de paso fraccionado --> Normalización por lotes --> Dropout, para prevenir el sobreajuste u overfitting, se aplica a los 3 primeros bloques --> ReLU.

Se aplica una convolución, después de la última capa, en el decodificador para mapear el número de canales de salida (el valor de OUTPUT_CHANNELS se establece en 3 en las redes creadas para imágenes a color y 1 en las de escala de grises), seguido de una función de activación Tanh (tangente hiperbólica). La función de activación Tanh solo se utiliza en la última convolución, debido a que resulta más pesada computacionalmente que las funciones ReLU.

```
last = tf.keras.layers.Conv2DTranspose(OUTPUT_CHANNELS, 4,  
                                       strides=2,  
                                       padding='same',  
                                       kernel_initializer=initializer,  
                                       activation='tanh')
```

Figura 2.32. Detalle de la aplicación de la función Tanh de activación después de la aplicación de la convolución en un fragmento de código perteneciente al modelo generador desarrollado
Fuente: elaboración propia

La normalización por lotes no se aplica a la primera capa del C64 en el codificador. Todas las funciones de activación ReLU del codificador tienen fugas, las ReLU del codificador tienen una pendiente de 0,2 (por lo tanto son LeakyReLU con un valor de α de 0,2), mientras que las ReLUs del decodificador no tienen fugas. Estas funciones de activación son las que incorporan la suma ponderada de todas las entradas de la capa anterior y generan un valor de resultado que pasa a la siguiente capa.

Como ya se ha mencionado, la arquitectura U-Net es idéntica a la arquitectura clásica del modelo Codificador-Decodificador, excepto en las conexiones de salto entre cada capa i en el codificador y la capa $n-i$ en el decodificador, donde n es el número total de capas. Las conexiones de salto concatenan las activaciones de la capa i a la capa $n-i$.

Un ejemplo de entrada podría ser una imagen (en blanco y negro), y la salida de esa imagen sería una versión en color.



Figura 2.33. Ejemplo de colorización de imagen con el modelo DeepMapsColorMultizoom
Fuente: elaboración propia, a partir de datos del IGN

El modelo del generador utilizado en el entrenamiento de las inteligencias artificiales ha sido programado en Python mediante las librerías Keras y TensorFlow, siguiendo la arquitectura descrita en este subepígrafe, la cual se

corresponde con la arquitectura del modelo generador de Pix2Pix propuesta por Isola et al. en su publicación de 2016 “Image-to-Image Translation with Conditional Adversarial Networks”.

Cabe mencionar lo mucho que ha simplificado las tareas de desarrollo de redes neuronales la librería Keras junto con TensorFlow, gracias a las cuales el generador U-Net, utilizado en los modelos generadores creados para esta investigación, puede implementarse con sólo unas pocas líneas de código que definan las distintas capas convolucionales y la función de activación, tal y como se muestra en la siguiente imagen.

```

def Generator():
    inputs = tf.keras.layers.Input(shape=[256,256,3])

    down_stack = [
        downsample(64, 4, apply_batchnorm=False),
        downsample(128, 4),
        downsample(256, 4),
        downsample(512, 4),
        downsample(512, 4),
        downsample(512, 4),
        downsample(512, 4),
        downsample(512, 4),
    ]

    up_stack = [
        upsample(512, 4, apply_dropout=True),
        upsample(512, 4, apply_dropout=True),
        upsample(512, 4, apply_dropout=True),
        upsample(512, 4),
        upsample(256, 4),
        upsample(128, 4),
        upsample(64, 4),
    ]

    initializer = tf.random_normal_initializer(0., 0.02)
    last = tf.keras.layers.Conv2DTranspose(OUTPUT_CHANNELS, 4,
                                           strides=2,
                                           padding='same',
                                           kernel_initializer=initializer,
                                           activation='tanh')

    x = inputs

    skips = []
    for down in down_stack:
        x = down(x)
        skips.append(x)

    skips = reversed(skips[:-1])
    for up, skip in zip(up_stack, skips):
        x = up(x)
        x = tf.keras.layers.Concatenate()([x, skip])

    x = last(x)

    return tf.keras.Model(inputs=inputs, outputs=x)

```

Figura 2.34. Modelo generador de tipo U-Net utilizado implementado en Python usando Keras
Fuente: elaboración propia

A continuación, se muestra la estructura del modelo generador U-Net implementado.

Layer (type)	Output Shape	Param #	Connected to
=====			
input_2 (InputLayer)	[(None, 256, 256, 3) 0		
sequential_20 (Sequential)	(None, 128, 128, 64) 3072		input_2[0][0]
sequential_21 (Sequential)	(None, 64, 64, 128) 131584		sequential_20[0][0]
sequential_22 (Sequential)	(None, 32, 32, 256) 525312		sequential_21[0][0]
sequential_23 (Sequential)	(None, 16, 16, 512) 2099200		sequential_22[0][0]
sequential_24 (Sequential)	(None, 8, 8, 512) 4196352		sequential_23[0][0]
sequential_25 (Sequential)	(None, 4, 4, 512) 4196352		sequential_24[0][0]
sequential_26 (Sequential)	(None, 2, 2, 512) 4196352		sequential_25[0][0]
sequential_27 (Sequential)	(None, 1, 1, 512) 4196352		sequential_26[0][0]
sequential_28 (Sequential)	(None, 2, 2, 512) 4196352		sequential_27[0][0]
concatenate_8 (Concatenate)	(None, 2, 2, 1024) 0		sequential_28[0][0] sequential_26[0][0]
sequential_29 (Sequential)	(None, 4, 4, 512) 8390656		concatenate_8[0][0]
concatenate_9 (Concatenate)	(None, 4, 4, 1024) 0		sequential_29[0][0] sequential_25[0][0]
sequential_30 (Sequential)	(None, 8, 8, 512) 8390656		concatenate_9[0][0]
concatenate_10 (Concatenate)	(None, 8, 8, 1024) 0		sequential_30[0][0] sequential_24[0][0]
sequential_31 (Sequential)	(None, 16, 16, 512) 8390656		concatenate_10[0][0]
concatenate_11 (Concatenate)	(None, 16, 16, 1024) 0		sequential_31[0][0] sequential_23[0][0]
sequential_32 (Sequential)	(None, 32, 32, 256) 4195328		concatenate_11[0][0]
concatenate_12 (Concatenate)	(None, 32, 32, 512) 0		sequential_32[0][0] sequential_22[0][0]
sequential_33 (Sequential)	(None, 64, 64, 128) 1049088		concatenate_12[0][0]
concatenate_13 (Concatenate)	(None, 64, 64, 256) 0		sequential_33[0][0] sequential_21[0][0]
sequential_34 (Sequential)	(None, 128, 128, 64) 262400		concatenate_13[0][0]
concatenate_14 (Concatenate)	(None, 128, 128, 128) 0		sequential_34[0][0] sequential_20[0][0]
conv2d_transpose_16 (Conv2DTran	(None, 256, 256, 3) 6147		concatenate_14[0][0]
=====			
Total params: 54,425,859			
Trainable params: 54,414,979			
Non-trainable params: 10,880			

Figura 2.35. Estructura del modelo generador U-Net utilizado
Fuente: elaboración propia

Para facilitar la comprensión de estos datos referentes al modelo generador implementado, puede recurrirse a mostrar los datos en forma de esquema mediante la función `plot_model()`, incluida en Keras.

```
tf.keras.utils.plot_model(generator, show_shapes=True, dpi=64)
```

Figura 2.36. Llamada a la función `plot_model()`
Fuente: elaboración propia

Mediante esta función se han generado los esquemas de los modelos generador y discriminador que se muestran a continuación.

Esquema de la arquitectura del generador

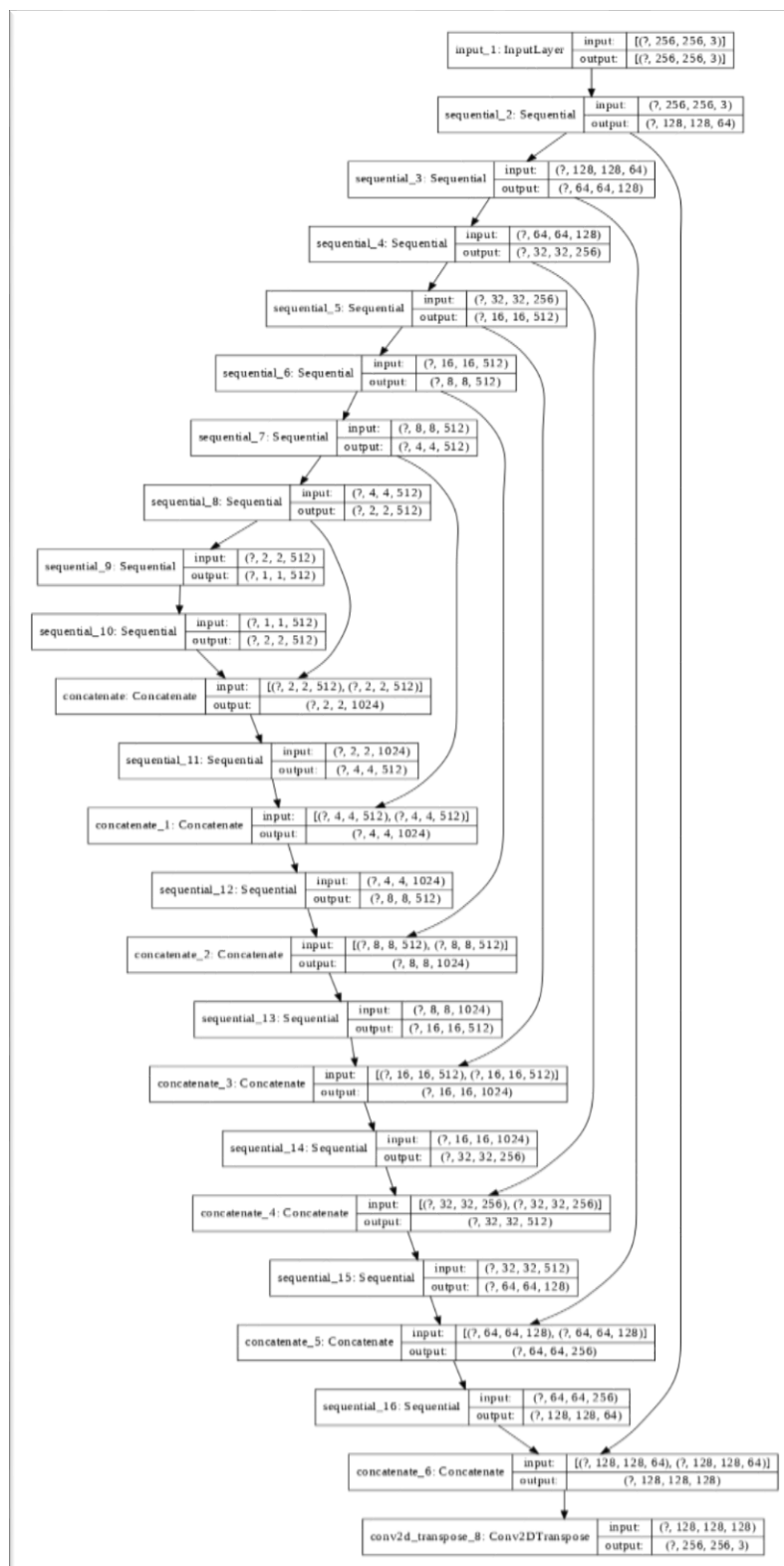


Figura 2.37. Esquema del modelo generador utilizado para el entrenamiento de las inteligencias artificiales y para su posterior aplicación
Fuente: elaboración propia

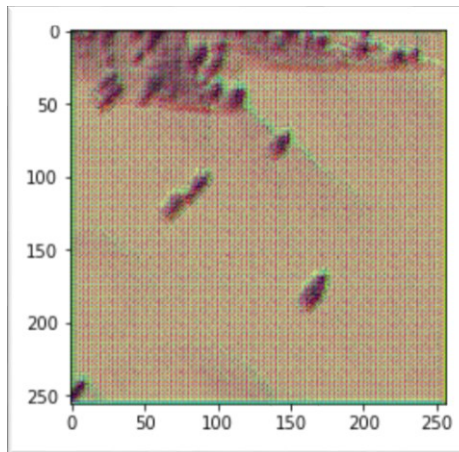
Pérdida del generador:

Figura 2.38. Imagen generada con Matplotlib a partir de los datos devueltos por el modelo generador antes de ser ajustado

Fuente: elaboración propia, a partir de datos del IGN

Los datos que componen la imagen anterior han sido generados por el modelo generador, antes de ser ajustado con los datos de entrenamiento y sin haber aplicado el optimizador basado en los gradientes derivados de los valores totales de pérdida del generador y del discriminador. Los valores representados en la figura 2.38 son los que van a utilizarse en la función de pérdida del generador. Los valores totales de pérdida del generador son calculados mediante la función de pérdida descrita en los siguientes párrafos. Para mostrar estos datos mediante Matplotlib se han escalado los valores del resultado a un rango válido para mostrarlos en RGB, dando valores entre 0 y 1 para los de tipo float o de coma flotante y de 1 a 255 a los enteros. Para la generación de la figura 2.38 se ha utilizado el siguiente código.

```
gen_output = generator(inp[tf.newaxis,...], training=False)
plt.imshow(gen_output[0,...])
```

Figura 2.39. Código para representar los valores devueltos por el modelo generador

Fuente: elaboración propia

La pérdida del generador es una pérdida de entropía cruzada sigmoidea de las imágenes generadas y una matriz de 256x256, formada en su totalidad por valores de 1, introducida en forma de array.

Para el cálculo de la pérdida total del generador, Isola y su equipo, en su publicación de 2016 “Image-to-Image Translation with Conditional Adversarial Networks”, incluyen una pérdida L1, la cual representa el error absoluto medio entre la imagen generada y la imagen real utilizada como referencia. Esta pérdida L1 permite que la imagen generada sea estructuralmente similar a la imagen de referencia. La fórmula para calcular la pérdida total del generador es pérdida del generador + LAMBDA * pérdida L1.

El hiperparámetro LAMBDA se ha establecido en valores que van del 5 al 100, dependiendo de la función que se busque al entrenar las inteligencias artificiales, siendo este valor de 100 el propuesto en la publicación antes mencionada.

La pérdida del generador influye en si el modelo generador puede producir imágenes que sean plausibles en el dominio de destino, mientras que la pérdida L1 regula el modelo generador para producir imágenes que sean una traducción plausible de la imagen de origen. Así pues, la combinación de la pérdida L1 con la pérdida del generador está controlada por LAMBDA. Por ejemplo, si se fija LAMBDA en 5, se dará 5 veces más importancia a la pérdida L1 que a la pérdida del generador durante el entrenamiento. Es decir, un valor de LAMBDA mayor produce resultados más fieles a la configuración de píxeles presente en la imagen de entrada, mientras que valores más bajos producen resultados más verosímiles en relación con el dominio de imágenes con las que es entrenada la inteligencia artificial. Por lo tanto, con valores de LAMBDA altos el resultado se parece más a la imagen de entrada y con valores más bajos se parece más a las imágenes utilizadas como modelos para el entrenamiento.

La función de pérdida del generador se ha implementado en Python siguiendo los criterios anteriormente descritos, tal y como se muestra en la siguiente imagen.

```
LAMBDA = 5
def generator_loss(disc_generated_output, gen_output, target):
    gan_loss = loss_object(tf.ones_like(disc_generated_output), disc_generated_output)

    l1_loss = tf.reduce_mean(tf.abs(target - gen_output))
    total_gen_loss = gan_loss + (LAMBDA * l1_loss)
    return total_gen_loss, gan_loss, l1_loss
```

Figura 2.40. Código Python de la función de pérdida del generador
Fuente: elaboración propia

2.3.2. Discriminador

A diferencia del modelo tradicional GAN, que utiliza una red neuronal convolucional profunda para clasificar las imágenes en el modelo discriminador, el modelo Pix2Pix utiliza como discriminador un modelo PatchGAN. PatchGAN es una red neuronal convolucional profunda, diseñada para clasificar parches de una imagen de entrada como real o falsa, en lugar de la imagen completa, lo cual produce un ahorro significativo de recursos y aumenta por lo tanto la velocidad de entrenamiento, sin que prácticamente se produzca ninguna merma en los resultados.

PatchGAN toma una imagen del dominio de origen y una imagen del dominio de destino y predice la probabilidad de que la imagen del dominio de destino sea una versión real o generada de la imagen de origen. La salida que produce es la probabilidad de que la imagen del dominio de destino sea una versión real de la imagen de origen.

La entrada al modelo discriminador resalta la necesidad de tener un conjunto de datos de imagen compuesto de imágenes de origen y de destino emparejadas cuando se entrena el modelo, tal y como se muestra a continuación con uno de los archivos de pares de imagen generados para el entrenamiento del modelo DeepMapsColor19.



Figura 2.41. Ejemplo de par de imágenes de entrenamiento
Fuente: elaboración propia, a partir de datos del IGN

El modelo discriminador PatchGAN se implementa como una red neuronal convolucional profunda, pero el número de capas se configura de tal manera que el campo receptivo efectivo de cada salida de la red se mapea a un tamaño específico en la imagen de entrada. La salida de la red es un mapa de predicciones de una sola característica (reales/falsas) que puede ser promediado para dar una sola puntuación.

Un tamaño de parche de 70×70 es efectivo en un amplio rango de tareas de traducción de imagen a imagen, por lo que la forma de la salida después de la última capa es (batch_size, 30, 30, 1). Cada parche de 30×30 de la salida clasifica una porción de 70×70 de la imagen de entrada. De esta forma, cada píxel de la imagen de 30×30 corresponde a la verosimilitud de un parche de 70×70 de la imagen de entrada (los parches se superponen mucho, ya que las imágenes de entrada son de 256×256). Este método de análisis por parches resulta mucho más eficiente que análisis tradicionales de la imagen completa, ya que consume muchos menos recursos computacionales y los resultados obtenidos son prácticamente iguales.

Cada bloque del discriminador está formado por los siguientes procesos (convolución --> normalización por lotes --> rectificador Leaky ReLU), se aplica una convolución después de la última capa para mapear a una salida unidimensional, seguida de una función de activación sigmoide del rectificador. La normalización por lotes no se aplica a la primera capa (C64). Todas las funciones de activación ReLU tienen fugas, con una pendiente de 0,2 (LeakyReLU). Todos los demás discriminadores siguen la misma arquitectura básica, con profundidad variada para modificar el tamaño del campo receptivo. Puede pensarse en la normalización por lotes como hacer un preprocesamiento en cada capa de la red.

Esquema de la arquitectura PatchGAN del discriminador:

Al igual que ocurre con la arquitectura del modelo del generador de la red, podemos generar un esquema de la arquitectura del modelo del discriminador mediante la función `plot_model()` de Keras, a través de la cual se ha generado el siguiente esquema.

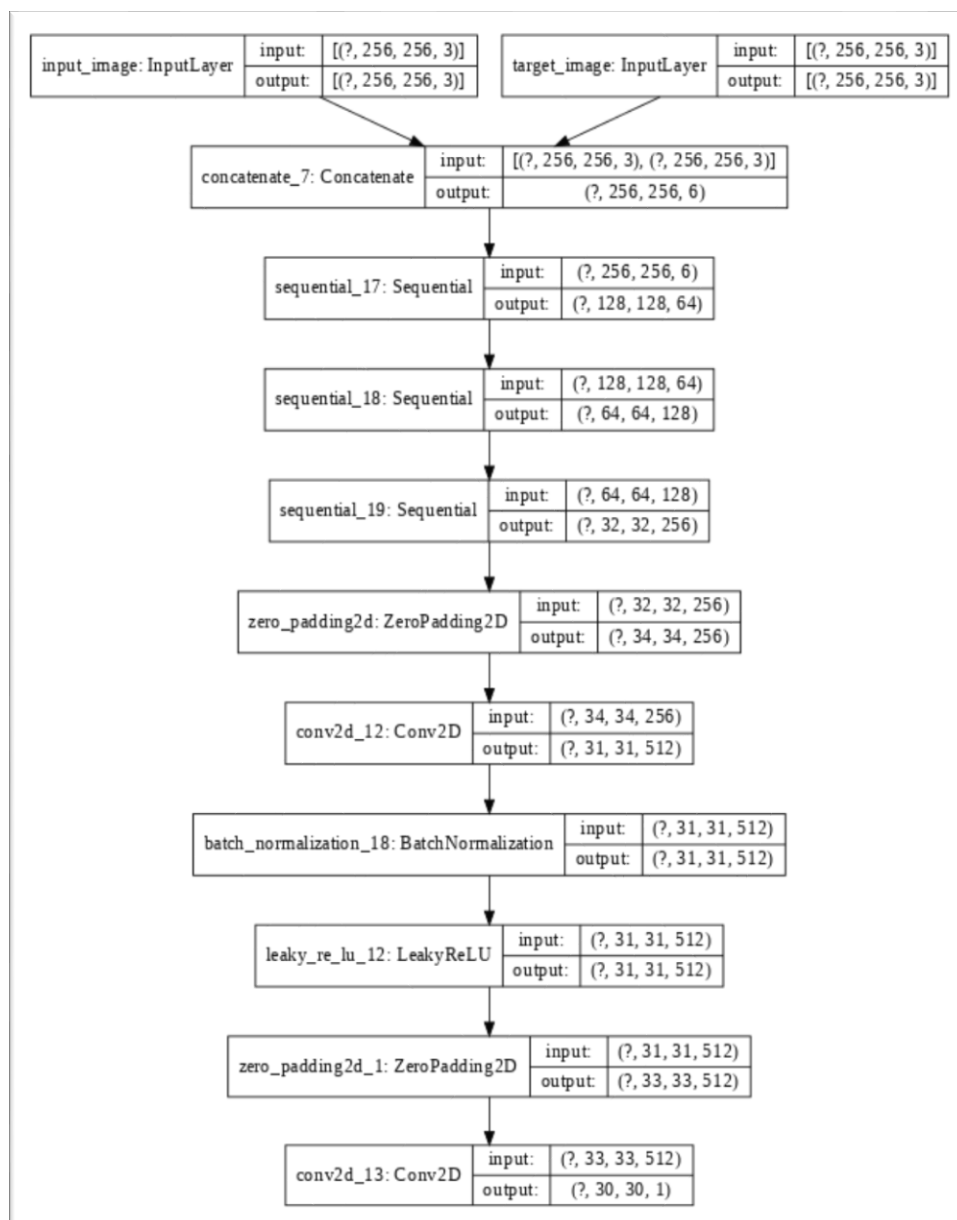


Figura 2.42. Esquema del modelo discriminador utilizado para el entrenamiento de las inteligencias artificiales
Fuente: elaboración propia

```
discriminator = Discriminator()  
tf.keras.utils.plot_model(discriminator, show_shapes=True, dpi=64)
```

Figura 2.43. Función para generar un esquema del modelo discriminador implementado anteriormente en el código
Fuente: elaboración propia

Cabe mencionar que los modelos VGG no tienen una batchnorm o capa de normalización por lotes, porque la normalización de lotes no existía en el momento en el que fue creada. Si lo entrenamos con él desde el principio, el peso preentrenado se beneficiará de la normalización de las activaciones. Así es que, añadir una capa de normalización por lotes mejora la eficiencia computacional de los modelos aquí desarrollados con respecto a otros modelos de tratamiento de imagen basados en VGG.

Si insertamos una normalización por lotes en una red preentrenada cambiará los pesos preentrenados, porque restará la media y se dividirá por la desviación estándar para las capas de activación y no queremos que eso suceda, ya que necesitamos que esos pesos preentrenados se mantengan iguales; es por ello por lo que se ha recurrido a entrenar las redes desde cero, en lugar de partir de modelos preentrenados y adaptarlos a nuestros propósitos.

El modelo del discriminador utilizado en el entrenamiento de las inteligencias artificiales ha sido programado en Python mediante las librerías Keras y TensorFlow, siguiendo la arquitectura descrita en este subepígrafe, la cual se corresponde con la arquitectura del modelo discriminador de Pix2Pix propuesta por Isola et al. en su publicación de 2016 “Image-to-Image Translation with Conditional Adversarial Networks”.


```
def Discriminator():
    initializer = tf.random_normal_initializer(0., 0.02)

    inp = tf.keras.layers.Input(shape=[256, 256, 3], name='input_image')
    tar = tf.keras.layers.Input(shape=[256, 256, 3], name='target_image')

    x = tf.keras.layers.concatenate([inp, tar])

    down1 = downsample(64, 4, False)(x)
    down2 = downsample(128, 4)(down1)
    down3 = downsample(256, 4)(down2)

    zero_pad1 = tf.keras.layers.ZeroPadding2D()(down3)
    conv = tf.keras.layers.Conv2D(512, 4, strides=1,
                                   kernel_initializer=initializer,
                                   use_bias=False)(zero_pad1)

    batchnorm1 = tf.keras.layers.BatchNormalization()(conv)

    leaky_relu = tf.keras.layers.LeakyReLU()(batchnorm1)

    zero_pad2 = tf.keras.layers.ZeroPadding2D()(leaky_relu)

    last = tf.keras.layers.Conv2D(1, 4, strides=1,
                                   kernel_initializer=initializer)(zero_pad2)

    return tf.keras.Model(inputs=[inp, tar], outputs=last)
```

Figura 2.44. Ejemplo del código que define la función del modelo discriminador
Fuente: elaboración propia

Al igual que ocurría con el generador, la implementación de la función del discriminador resulta igualmente sencilla gracias a la librería Keras, pudiendo definirse en sólo unas cuantas líneas.

Pérdida del discriminador

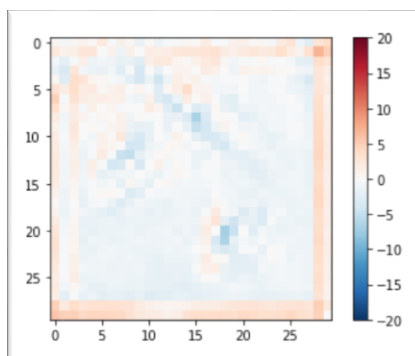


Figura 2.45. Heatmap generado a partir de los resultados devueltos por el modelo discriminador sin haber sido ajustado por los datos del entrenamiento, al ser aplicado a una imagen real y la imagen generada por el modelo generador
Fuente: elaboración propia

La imagen mostrada anteriormente ha sido generada mediante la librería Matplotlib, a través de la cual se ha creado un gráfico que representa los valores devueltos como resultado de la aplicación del modelo discriminador, antes de ser ajustado con los datos de entrenamiento y sin haber aplicado el optimizador basado en los gradientes derivados de los valores totales de pérdida del discriminador y del generador. Los valores representados en la figura 2.45 van a utilizarse en la función de pérdida del discriminador. Los valores totales de pérdida del discriminador son calculados mediante la función de pérdida descrita en los siguientes párrafos.

El gráfico mostrado en la figura 2.45 se ha calculado mediante el uso de un par de imágenes de entrenamiento, el cual incluye la imagen real o imagen modelo y la imagen que se va a utilizar como entrada para el modelo generador.

Para la generación del gráfico, se ha calculado el valor de cada píxel de la imagen mostrada a partir de cada píxel de una imagen modelo (original) en relación con los píxeles correspondientes de la imagen generada con el modelo generador a partir de la imagen de entrada, ajustando los resultados a una escala entre -20 y 20 y visualizándolos mediante una paleta de colores que va del azul al rojo, siendo el color azul el correspondiente a los valores de píxel de -20 y el rojo a los de 20. Para mostrar estos datos mediante Matplotlib se han escalado los valores del resultado a un rango válido para mostrarlos en RGB, dando valores entre 0 y 1 para los de tipo float o de coma flotante y de 1 a 255 a los enteros. Este cálculo de las discrepancias entre la imagen original y la generada por el modelo generador a partir de la imagen de entrada es crucial para establecer los valores de las funciones de pérdida y por lo tanto los pesos, lo cual es la base del aprendizaje de las inteligencias artificiales. Para la generación de la figura 2.45 se ha utilizado el código mostrado en la figura 2.46.

```
disc_out = discriminator([inp[tf.newaxis,...], gen_output], training=False)
plt.imshow(disc_out[0,...,-1], vmin=-20, vmax=20, cmap='RdBu_r')
plt.colorbar()
```

Figura 2.46. Ejemplo de código Python utilizado para generar una imagen del resultado de la aplicación del modelo discriminador sin ajustar y alimentado con una imagen real y la generada con el modelo generador

Fuente: elaboración propia

El modelo discriminador toma 2 entradas, las imágenes reales y las imágenes generadas.

La pérdida real es una pérdida de entropía cruzada sigmoidea de las imágenes reales y una matriz de unos (ya que éstas son las imágenes reales), mientras que la pérdida generada es una pérdida de entropía cruzada sigmoidea de las imágenes generadas y una matriz de ceros (ya que éstas son las imágenes falsas).

En base a los valores de la pérdida real y la pérdida generada, se calcula la pérdida total, que es la suma de la pérdida real y la pérdida generada.

Función del total de pérdida del discriminador = pérdida real + pérdida generada

Al igual que con el resto de las funciones, la librería Keras nos facilita en gran medida el desarrollo de la función de pérdida del discriminador, cuyo código en Python se corresponde con los criterios descritos anteriormente y puede expresarse tal y como se muestra en la siguiente imagen.

```

loss_object = tf.keras.losses.BinaryCrossentropy(from_logits=True)
def discriminator_loss(disc_real_output, disc_generated_output):
    real_loss = loss_object(tf.ones_like(disc_real_output), disc_real_output)

    generated_loss = loss_object(tf.zeros_like(disc_generated_output), disc_generated_output)

    total_disc_loss = real_loss + generated_loss

    return total disc loss

```

Figura 2.47. Ejemplo del código utilizado para describir la función de pérdida del modelo discriminador

Fuente: elaboración propia

El cálculo de los gradientes basados en los valores de pérdida tanto del generador como del discriminador es importante para que éstos puedan ser aplicados en los optimizadores del generador y del discriminador, los cuales mejoran el proceso de entrenamiento y los resultados obtenidos.

A continuación, se muestra la función de iteración del entrenamiento, donde se aprecia la aplicación de los gradientes anteriormente mencionados en cada una de las iteraciones del entrenamiento.

```

def train_step(input_image, target, epoch):
    with tf.GradientTape() as gen_tape, tf.GradientTape() as disc_tape:
        gen_output = generator(input_image, training=True)
        disc_real_output = discriminator([input_image, target], training=True)
        disc_generated_output = discriminator([input_image, gen_output], training=True)
        gen_total_loss, gen_gan_loss, gen_l1_loss = generator_loss(disc_generated_output, gen_output, target)
        disc_loss = discriminator_loss(disc_real_output, disc_generated_output)
        generator_gradients = gen_tape.gradient(gen_total_loss, generator.trainable_variables)
        discriminator_gradients = disc_tape.gradient(disc_loss, discriminator.trainable_variables)
        generator_optimizer.apply_gradients(zip(generator_gradients, generator.trainable_variables))
        discriminator_optimizer.apply_gradients(zip(discriminator_gradients, discriminator.trainable_variables))
    with summary_writer.as_default():
        tf.summary.scalar('gen_total_loss', gen_total_loss, step=epoch)
        tf.summary.scalar('gen_gan_loss', gen_gan_loss, step=epoch)
        tf.summary.scalar('gen_l1_loss', gen_l1_loss, step=epoch)
        tf.summary.scalar('disc_loss', disc_loss, step=epoch)

```

Figura 2.48. Código Python de la función de entrenamiento

Fuente: elaboración propia

2.3.3. Entrenamiento

Para entrenar la red hay que dar dos pasos: entrenar al discriminador y entrenar al generador. Todas las redes fueron entrenadas desde cero. Los pesos se inicializan a partir de una distribución gaussiana con media 0 y desviación estándar de 0.02; el proceso se realiza de forma sencilla gracias a la librería TensorFlow, tal y como se muestra a continuación.

```
initializer = tf.random_normal_initializer(0., 0.02)
```

Figura 2.49. Inicialización de pesos
Fuente: elaboración propia

Como ya se ha mencionado anteriormente, para entrenar los modelos se ha recurrido a teselas de 256x256, aunque también se han realizado pruebas con otras dimensiones en los ficheros de imagen. Para la ingesta de los archivos utilizados en el entrenamiento se han unido los pares de imágenes de entrada y de imagen real o imagen modelo, al procesar cada uno de estos archivos de 512x256 píxeles; el script desarrollado para el entrenamiento divide al archivo ingestado separando las dos teselas de 256x256 de cada ejemplo. Para ello se utiliza una función de carga de las imágenes de entrenamiento que realiza dicha división.

```
def load(image_file):  
    image = tf.io.read_file(image_file)  
    image = tf.image.decode_jpeg(image)  
    w = tf.shape(image)[1]  
    w = w // 2  
    real_image = image[:, :w, :]  
    input_image = image[:, w:, :]  
    input_image = tf.cast(input_image, tf.float32)  
    real_image = tf.cast(real_image, tf.float32)  
    return input_image, real_image
```

Figura 2.50. Función de carga de imágenes de ejemplo
Fuente: elaboración propia

En el entrenamiento de las inteligencias artificiales desarrolladas se ha recurrido a algunas técnicas que mejoran el rendimiento de los entrenamientos, ya habituales en el campo del entrenamiento de redes convolucionales. Una de estas técnicas es el redimensionado de los archivos que se utilizan en el entrenamiento y el posterior recorte de parte de la nueva imagen redimensionada. Gracias a este procedimiento se consigue generar una mayor variedad de ejemplos partiendo de los mismos datos originales.

El redimensionado consiste en escalar cada una de las teselas de 256x256 píxeles a imágenes de 286x286 píxeles. Para redimensionar las imágenes se utiliza la interpolación del vecino más próximo o nearest neighbor. Para facilitar el proceso de redimensionado de imágenes se ha utilizado la siguiente función, en la cual se pueden ajustar los valores de ancho y alto de la imagen en función de lo que se desee; en nuestro caso, al realizar la llamada a la función, estos parámetros se ajustan a 286x286 píxeles.

```
def resize(input_image, real_image, height, width):  
    input_image = tf.image.resize(input_image, [height, width],  
                                method=tf.image.ResizeMethod.NEAREST_NEIGHBOR)  
    real_image = tf.image.resize(real_image, [height, width],  
                                method=tf.image.ResizeMethod.NEAREST_NEIGHBOR)  
  
    return input_image, real_image
```

Figura 2.51. Función de reescalado de imágenes
Fuente: elaboración propia

A continuación, como ya se ha mencionado anteriormente, en cada iteración del entrenamiento se corta una tesela de 256x256 píxeles correspondiente a una posición aleatoria de la original reescalada a 286x286 píxeles. Para ello se utiliza la siguiente función.

```
def random_crop(input_image, real_image):  
    stacked_image = tf.stack([input_image, real_image], axis=0)  
    cropped_image = tf.image.random_crop(  
        stacked_image, size=[2, IMG_HEIGHT, IMG_WIDTH, 3])  
  
    return cropped_image[0], cropped_image[1]
```

Figura 2.52. Función de corte aleatorio de teselas
Fuente: elaboración propia

Como puede apreciarse en el código anterior, los valores de alto y ancho del corte realizado vienen definidos por dos constantes, las cuales han sido definidas tras la carga de las librerías, junto con otras constantes utilizadas en el código.

```
BUFFER_SIZE = 600  
BATCH_SIZE = 1  
IMG_WIDTH = 256  
IMG_HEIGHT = 256
```

Figura 2.53. Definición de constantes
Fuente: elaboración propia

Además de recurrir al redimensionado y corte aleatorio de las teselas de entrenamiento, también se ha recurrido a otra técnica muy habitual en el entrenamiento de redes convolucionales, consistente en realizar una rotación aleatoria de las imágenes para, de esta manera, proporcionar una mayor variedad en el entrenamiento y que las inteligencias resultantes puedan actuar de forma correcta en la mayoría de los casos, identificando elementos independientemente de su orientación. Tanto el redimensionado, como el corte y la rotación aleatoria se realizan a través de la siguiente función, que a su vez llama a las funciones anteriormente definidas y las configura con los parámetros necesarios.

```
def random_jitter(input_image, real_image):  
    input_image, real_image = resize(input_image, real_image, 286, 286)  
    input_image, real_image = random_crop(input_image, real_image)  
    if tf.random.uniform(()) > 0.5:  
        input_image = tf.image.flip_left_right(input_image)  
        real_image = tf.image.flip_left_right(real_image)  
    return input_image, real_image
```

Figura 2.54. Función de redimensionamiento, corte y rotación aleatorios de las teselas
Fuente: elaboración propia

Gracias a las funciones anteriormente descritas, de una única tesela de ejemplo pueden generarse varios ejemplos para el entrenamiento, tal y como se muestra a continuación.

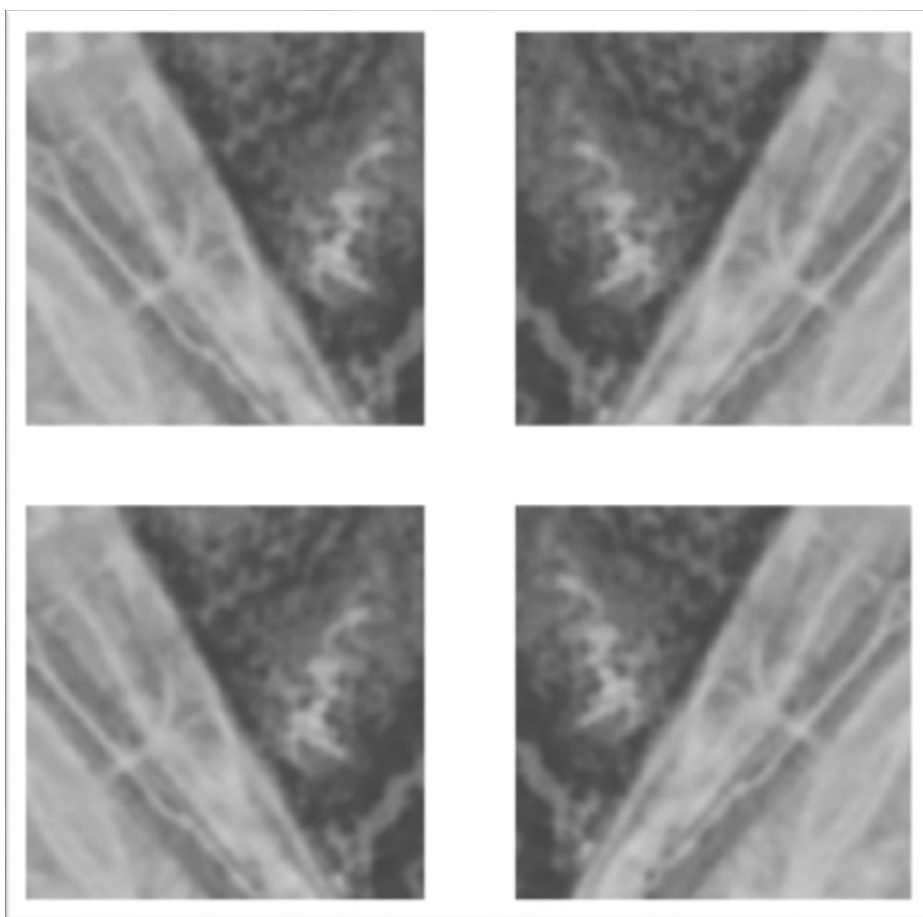


Figura 2.55. Variaciones de teselas realizadas a partir de la imagen original
Fuente: elaboración propia a partir de datos del IGN

A estas variaciones deben añadirse otras variaciones aplicadas, no descritas en la publicación de Isola y su equipo en 2016, como son la introducción de ruido y desenfoque, además de modificaciones de la luminosidad y el contraste para algunos de los modelos creados. Este tipo de variaciones, al ser aplicadas adecuadamente, mejoran en gran medida los resultados obtenidos en determinadas tareas. Posiblemente la mayor innovación introducida en el entrenamiento de determinados modelos aquí desarrollados es la creación de teselas derivadas de la original, en las que se ha modificado la luminosidad y el contraste.

Este tipo de modificaciones son fundamentales en el trabajo cartográfico, y debido a que hasta ahora no se habían aplicado las redes neuronales a las labores cartográficas expuestas en este documento, nadie se había planteado la necesidad de variar la luminosidad y el contraste en este procesado previo de las imágenes originales. Para poder trabajar con cartografías antiguas recogidas en medios analógicos es fundamental introducir variaciones de la luminosidad y el contraste en los datos de entrenamiento, puesto que las ortoimágenes antiguas presentan una gran variación de estos factores debido a la falta de estandarización de los resultados, derivada de los procesos analógicos de captura de imagen y revelado.

```
for i in range(0,8):
    for j in range(0,8):
        finalbucle = Image.new("RGB", (512,256), "black")
        brillo1= cv2.convertScaleAbs(np.float32(imagen1.convert('L')), alpha=1+j/10, beta=i*5)
        backtorgb = cv2.cvtColor(brillo1,cv2.COLOR_GRAY2RGB)
        finalbucle.paste( imagen1, (0,0))
        finalbucle.paste( im1, (256,0))
        finalbucle.save( "entrenamiento\\" + "originalcontraste_"+str(j)+"_brillo"+ str(i) +directorio2 + "_" + fname)
for i in range(0,8):
    for j in range(0,8):
        finalbucle = Image.new("RGB", (512,256), "black")
        brillo1= cv2.convertScaleAbs(np.float32(desenfocada2), alpha=1+j/10, beta=i*5)
        backtorgb = cv2.cvtColor(brillo1,cv2.COLOR_GRAY2RGB)
        finalbucle.paste( imagen1, (0,0))
        finalbucle.paste( im1, (256,0))
        finalbucle.save( "entrenamiento\\" + "des2contraste_"+str(j)+"_brillo"+ str(i) +directorio2 + "_" + fname)
```

Figura 2.56. Fragmento del código implementado para generar 64 variaciones de la imagen original con diferentes configuraciones de brillo y contraste

Fuente: elaboración propia

Una vez disponibles los archivos, el modelo generador, para cada imagen de entrada, genera una imagen de salida. El modelo discriminador recibe la imagen modelo (imagen original) y la imagen generada por el modelo generador como primera entrada. La segunda entrada es la imagen de entrada y la imagen modelo (imagen original). A continuación, se calculan la pérdida del generador y la pérdida del discriminador. Posteriormente, se calculan los gradientes de pérdida con respecto, tanto a la variable generadora como a la discriminadora, y se aplican al optimizador. Es decir, el generador genera una imagen de salida. El discriminador analiza el par de imágenes entrada/objetivo (imagen modelo original) y el par de entrada/salida, produciendo su conjetura sobre lo realistas que parecen. El vector de pesos del discriminador se ajusta entonces en base al error de clasificación del par entrada/salida y del par entrada/objetivo.

El modelo discriminador se entrena de forma autónoma, de la misma manera que un modelo de red generativa antagónica tradicional, minimizando la probabilidad de registro negativo al identificar imágenes reales y falsas, aunque condicionado a una imagen de origen.

El entrenamiento del discriminador es demasiado rápido comparado con el generador, por lo que la pérdida del discriminador se reduce a la mitad para ralentizar el aprendizaje del mismo y de esa forma equilibrar el proceso de entrenamiento.

$$\text{Pérdida del discriminador} = 0,5 * \text{Pérdida del discriminador}$$

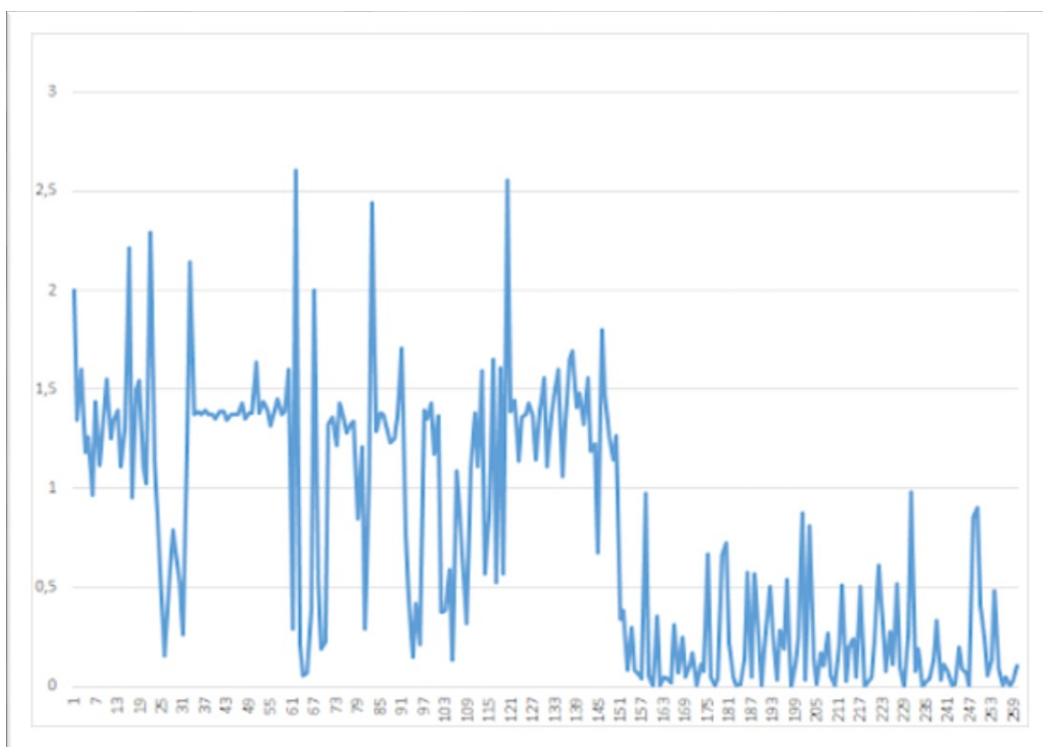


Figura 2.57. Valores de la función de pérdida del discriminador durante 260 iteraciones (modelo DeepMapsColor20)
Fuente: elaboración propia

El modelo generador se entrena utilizando tanto la pérdida de la opuesta para el modelo discriminador, como la L1 o diferencia media de píxeles absolutos entre la traducción generada de la imagen de origen y la imagen de destino esperada.

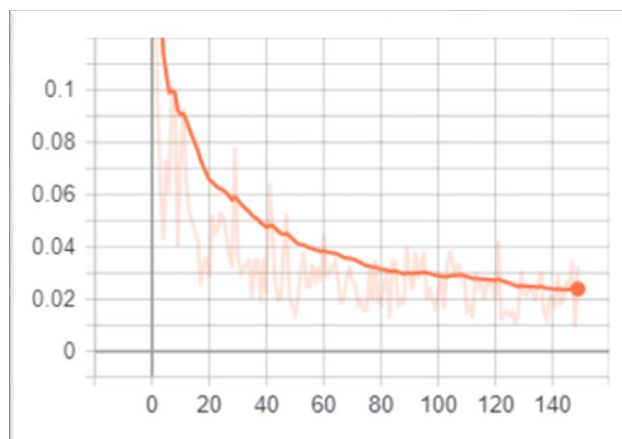


Figura 2.58. Valores de la función de pérdida L1 durante las primeras 151 iteraciones (modelo DeepMapsColor20)
Fuente: elaboración propia

La pérdida del generador y la pérdida L1 se combinan en una función de pérdida compuesta que da como resultado el valor total de pérdida del generador, que se utiliza para actualizar el modelo generador. La precisión del modelo generador durante el entrenamiento aumenta a medida que el valor de la función de pérdida total disminuye.

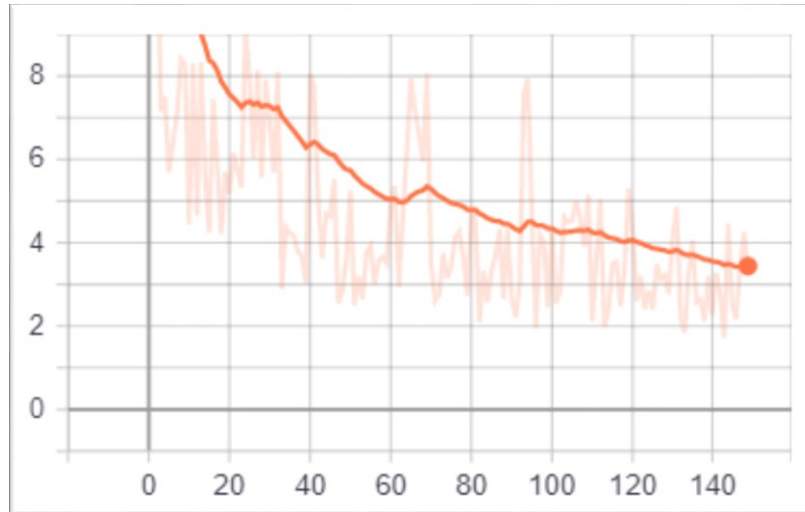


Figura 2.59. Valores de la función de pérdida total del generador durante las primeras 151 iteraciones (modelo DeepMapsColor20)

Fuente: elaboración propia

La pérdida adversarial influye en si el modelo generador puede producir imágenes que sean plausibles en el dominio de destino, mientras que la pérdida L1 regulariza el modelo generador para producir imágenes que sean una traducción plausible de la imagen de origen, es por ello por lo que, cuanto más disminuya la pérdida total del generador, mejores resultados devolverán las inteligencias artificiales. La combinación de la pérdida L1 con la pérdida adversarial está controlada por un hiperparámetro LAMBDA, que se fija normalmente en 100, aunque en algunos de los modelos creados se ha fijado en valores distintos, dando 100 (o el valor fijado para LAMBDA) veces más importancia a la pérdida L1 que a la pérdida adversarial para el generador durante el entrenamiento.

$$\text{Pérdida total del generador} = \text{pérdida adversarial} + \text{LAMBDA} * \text{Pérdida L1}$$

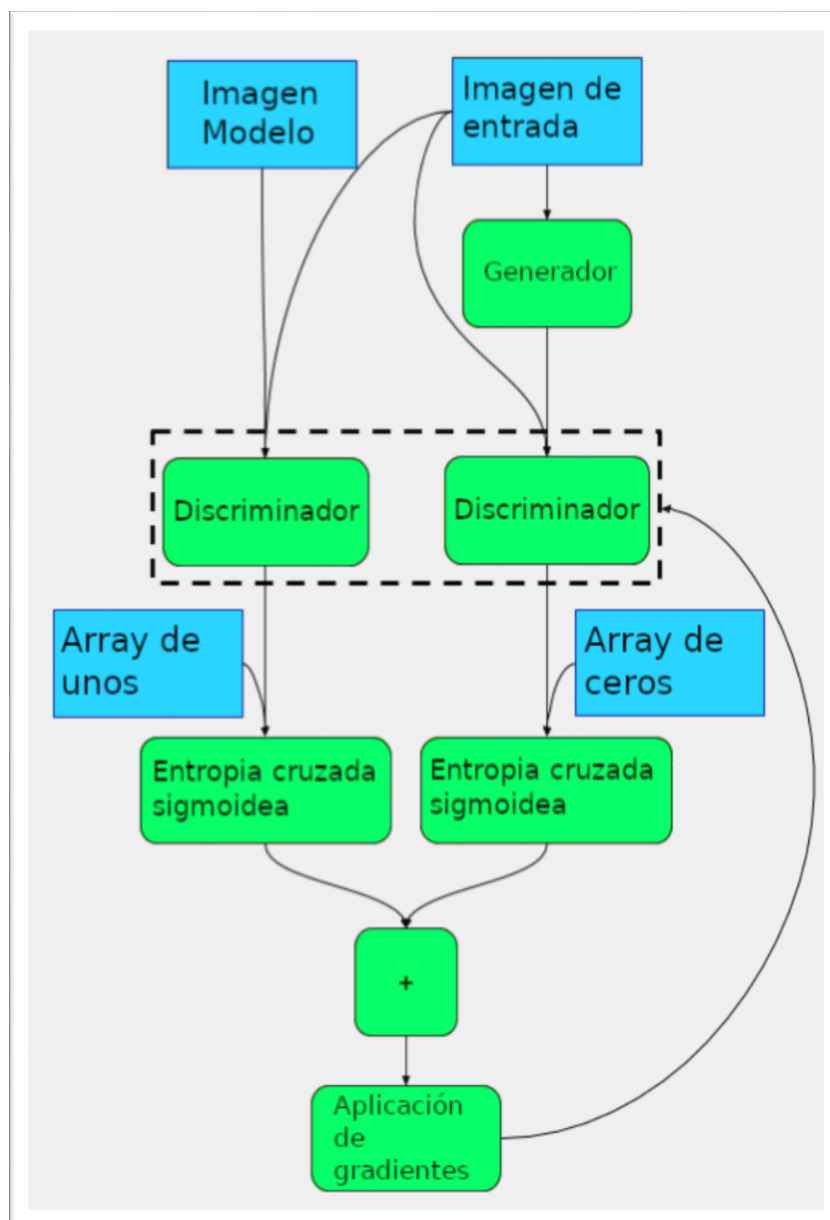
Esquema del entrenamiento del discriminador:

Figura 2.60. Esquema del entrenamiento del modelo discriminador
Fuente: elaboración propia

En primer lugar, el generador crea una imagen de salida. El discriminador analiza el par de entrada/objetivo y el par de entrada/salida, y realiza un juicio sobre lo realistas que parecen. El vector de pesos del discriminador se ajusta entonces en base al error de clasificación del par entrada/salida y del par entrada/destino.

Esquema de entrenamiento del generador:

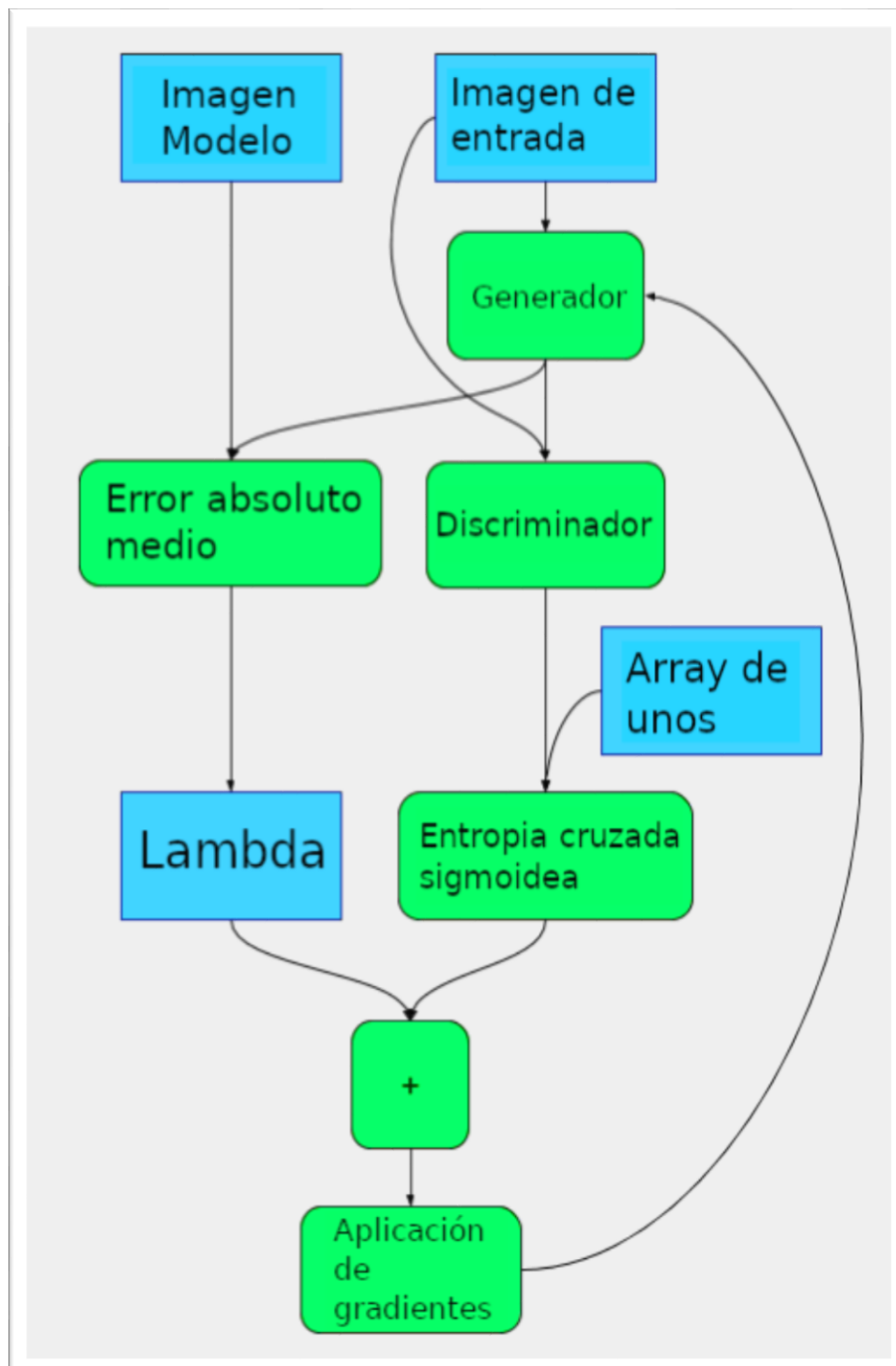


Figura 2.61 Esquema del entrenamiento del modelo generador
Fuente: elaboración propia

Los pesos del generador se ajustan entonces en base a la salida del discriminador, así como a la diferencia entre la salida y la imagen objetivo.

Como el generador se entrena en base a la salida del discriminador, en realidad, el gradiente se calcula a través del discriminador (Figura 2.59), lo que significa que, mientras el discriminador mejora, también se está entrenando al generador para vencer al discriminador. Si el discriminador es bueno en su trabajo y el generador es capaz de aprender la función correcta de mapeo a través del descenso del gradiente, el sistema producirá imágenes que deberían poder engañar a un humano.



Figura 2.62. Ejemplo de imagen generada con el modelo DeepMapsColor17 frente a la imagen modelo usada para el entrenamiento

Fuente: elaboración propia, a partir de datos del IGN

2.3.4. Evaluación del entrenamiento

Para evaluar los avances en el entrenamiento de las inteligencias artificiales, aparte de la revisión visual de los resultados, se ha utilizado la herramienta TensorBoard, de TensorFlow, para su seguimiento y visualización, obteniendo así las gráficas de precisión y pérdida.

La llamada a TensorBoard y la carga de los logs de entrenamiento puede realizarse de forma sencilla desde el código de entrenamiento, añadiendo un par de líneas en las que se incluye la ruta dónde se han guardado los registros de los entrenamientos, tal y como se muestra en la siguiente imagen.

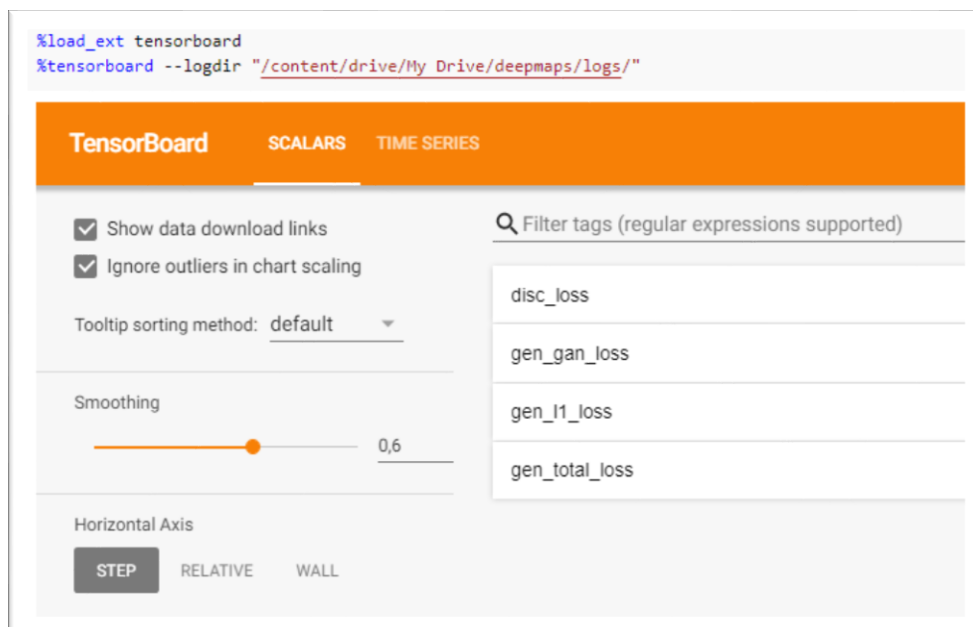


Figura 2.63. Ejemplo de carga de registros en TensorBoard
Fuente: elaboración propia

Los gráficos de los valores de pérdida expuestos en este capítulo han sido obtenidos cargando los logs de entrenamiento en TensorBoard. En el caso del gráfico de la pérdida del discriminador se han tenido que descargar los datos de varias sesiones de entrenamiento en CSV y graficarlos mediante Microsoft Excel. El motivo de esto es que, como ya se explicó anteriormente, las

limitaciones de uso de los servidores de GoogleColaboratory restringían el tiempo de uso continuado de sus máquinas, por lo que cada sesión de entrenamiento estaba compuesta por un número variable de iteraciones; por ello, para poder incluir las iteraciones de varias sesiones de entrenamiento en una misma gráfica, es necesario descargar los datos en bruto y unirlos para su posterior visualización. En la esquina inferior izquierda de la imagen mostrada a continuación, pueden verse las distintas sesiones de entrenamiento de un mismo modelo, en este caso DeepMapsRestaura15.

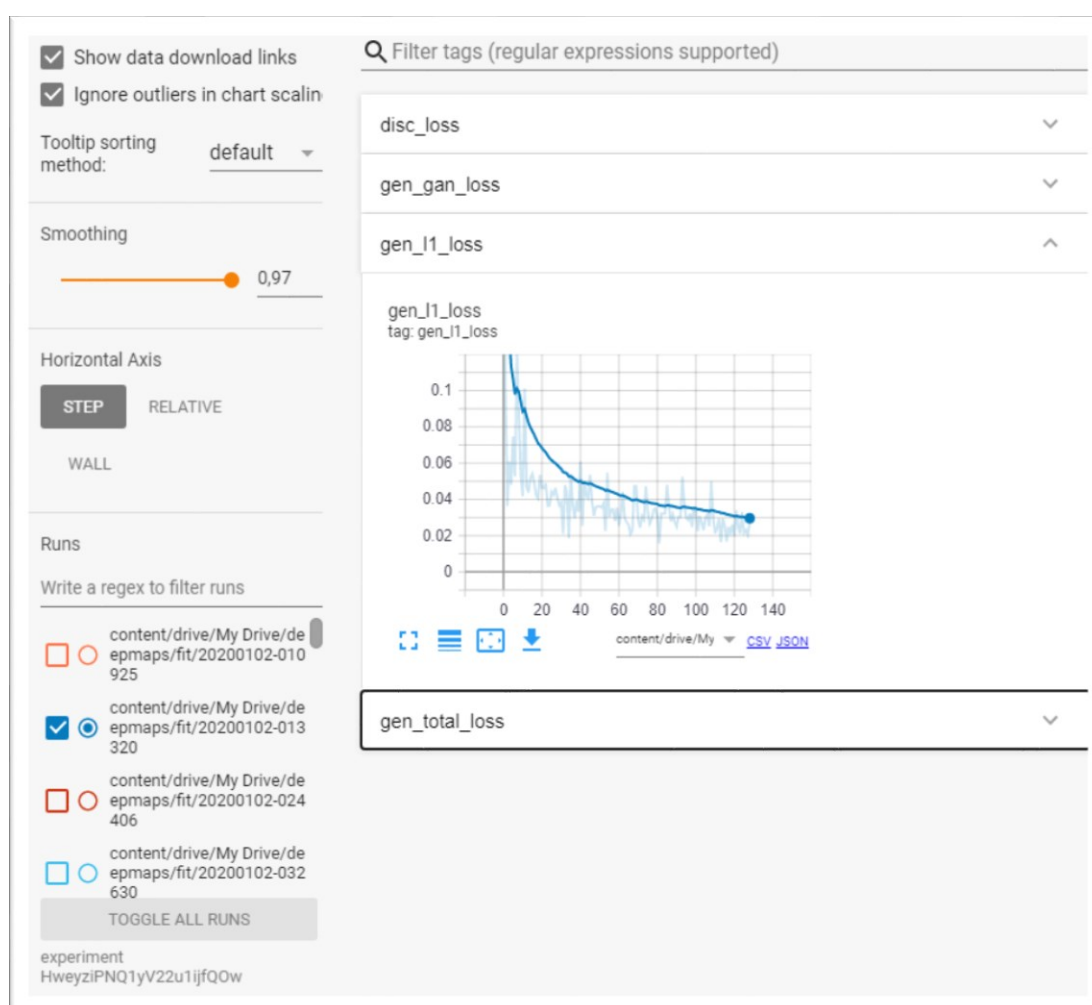


Figura 2.64. Ejemplo del panel de control de TensorBoard (Modelo DeepMapsRestaura15)
Fuente: elaboración propia

2.4. Librerías utilizadas

Para el desarrollo, tanto del código necesario para el entrenamiento de las inteligencias artificiales, como para su posterior aplicación, se ha recurrido a las siguientes librerías: TensorFlow, Keras, Numpy, Os, Time, ZipFile, Matplotlib, lpython.

Dentro de las librerías anteriormente mencionadas las más cruciales para la creación de las redes neuronales son TensorFlow, Keras y Numpy. Hay que matizar que en la versión utilizada de TensorFlow (2.3), las librerías Keras y Numpy vienen incluidas como módulos de esta. Por lo tanto, la librería más destacada para las redes neuronales creadas es TensorFlow, siendo el resto necesarias para realizar algunas operaciones de lectura, escritura y compresión de archivos, o generar la salida gráfica, como es el caso de Matplotlib.

2.5. Hardware utilizado en los entrenamientos

Para realizar los distintos entrenamientos, debido al coste computacional que supone el entrenamiento de redes profundas, se ha optado por el procesamiento distribuido en la nube mediante el uso de diferentes servidores, ya que el equipo personal no tendría capacidad suficiente para realizar todos los cálculos necesarios. Por este motivo, se ha realizado el entrenamiento de las distintas redes ejecutando los códigos, que han sido implementados durante estas investigaciones, de forma paralela en los servidores ofrecidos por Google Colaboratory.

Las características de los servidores en los que se ha entrenado cada red son las siguientes:

- **GPU:** 1xTesla K80, compute 3.7, having 2496 CUDA cores, 12GB GDDR5 VRAM
- **CPU:** 1xsingle core hyper threaded Xeon Processors @2.3Ghz i.e. (1 core, 2 threads)
- **RAM:** ~12.6 GB
- **Disco Duro:** ~64 GB

2.6. Metodología de trabajo para la aplicación práctica de redes neuronales en la cartografía

Debido a las limitaciones en cuanto a capacidad de computación que tienen los equipos actuales, para el tratamiento masivo de datos deben seguirse algunas de las pautas que ya son habituales en el trabajo con Big Data.

Mediante el procesamiento distribuido pueden resolverse parte de los problemas que surgen y que dificultan o, incluso, impiden el tratamiento masivo de los datos en un equipo local. Sin embargo, esto no supone la posibilidad de tener capacidad de computación ilimitada, ya que, por muchos recursos de los que se disponga, el número de máquinas disponibles para el procesamiento de los datos siempre va a ser finito; a esto hay que añadir que la comunicación entre las mismas también limita la velocidad y el acceso a sus capacidades de forma plena; en otras palabras 10 ordenadores no van a tener 10 veces más capacidad de computación que un ordenador.

Dependiendo de la tarea a desarrollar será necesario un número de servidores y un tiempo para que éstos computen la información. Lógicamente, a más servidores estén a nuestra disposición y más potentes sean éstos, el tiempo de cómputo será menor. No obstante, existen ciertas limitaciones a este principio general derivadas de las comunicaciones entre las máquinas destinadas al trabajo y de la asignación de tareas a cada una de ellas.

Independientemente del número de máquinas que se utilicen, el procesamiento de las imágenes consume recursos, y cuanto más complejo sea ese procesamiento mayor la cantidad de recursos consumidos. Lógicamente, para ahorrar trabajo a los humanos y poder llevar a cabo incluso tareas cuya realización sería imposible de forma manual, la tendencia es la automatización de procesos.

Hay que tener en cuenta la complejidad de las redes neuronales aquí desarrolladas y lo distintas que son de la mera aplicación de filtros tradicionales, basados en algoritmos simples. Para poder entender un poco mejor esta complejidad, cabe mencionar que los algoritmos de los modelos creados poseen varias decenas de millones de parámetros cada uno, y, por lo tanto, se acercan más a los procesos cognitivos biológicos de visión que a la simple aplicación de fórmulas matemáticas relativamente sencillas, desarrolladas por humanos, a los píxeles.

Los paralelismos entre las Redes Neuronales Biológicas implicadas en la visión y las Redes Neuronales Convolucionales, como las utilizadas en el generador y el discriminador de los modelos implementados, ya han sido analizados en múltiples investigaciones en los campos tanto de la neurología como de la informática; igual ocurre con los paralelismos entre los mecanismos biológicos de activación de las neuronas y las funciones de activación utilizadas en las redes neuronales artificiales, siendo sorprendentemente similares los umbrales de activación y la consolidación de vías neuronales que se dan, tanto en las redes biológicas como en las artificiales.

Debido a que la automatización de tareas complejas, como puede ser la restauración de ortoimágenes aéreas, aglutina un gran número de subtareas (dar color, corregir errores, mejorar el enfoque y la resolución, etc.), para cuya ejecución se consume un gran número de recursos, incluso teniendo en cuenta que la inteligencia artificial produce algoritmos mucho más eficientes que los equivalentes que pudiesen llegar a desarrollar los humanos, siguen siendo muy costosos como para ser aplicados a grandes imágenes, ya que cuantos más píxeles se deban procesar se necesita mayor capacidad de cálculo.

Este tipo de tareas complejas no pueden entenderse como la simple aplicación de un filtro o un algoritmo sencillo a una imagen, los cuales en muchos casos puede aplicarse a imágenes grandes debido a que se procesan los píxeles de forma secuencial (los algoritmos tradicionales no tienen en cuenta el

contexto). En el caso de labores más complejas no basta con analizar un píxel o su vecindad, sino que, al igual que sucede con el procesamiento de imágenes que ocurre de forma biológica, la comprensión de la imagen requiere el análisis de la totalidad de ésta, ya que los procesos cognitivos tanto humanos como artificiales necesitan un contexto. Esto obliga a procesar las imágenes como un todo, a pesar de que, en algunos de los procesos del entrenamiento, esta imagen total internamente se descompone en partes; es por ello por lo que las dimensiones de las imágenes a tratar son limitadas y las imágenes con un gran número de píxeles, como las utilizadas habitualmente en los datos ráster son demasiado grandes como para ser procesadas de una sola vez. Debido a esto, tanto para el entrenamiento como para el procesamiento de los datos, se ha recurrido a un recurso ya utilizado en cartografía digital, el uso de teselas. Este recurso permite procesar las imágenes en bloques más pequeños, los cuales pueden ser distribuidos a los distintos servidores encargados del procesamiento y abarcados por la inteligencia artificial de forma secuencial, pero lo suficientemente grandes como para que en ellos puedan distinguirse elementos y estructuras como árboles, ríos, edificios, etc.

El uso de teselas en cartografía vino impulsado por las limitaciones de memoria en los sistemas de visualización y la velocidad en las comunicaciones; es por ello que son tan utilizadas en la cartografía digital a través de la web (principalmente en datos ráster, aunque también se ha utilizado el fraccionado de modelos tridimensionales en el caso de los datos de relieve mundial utilizados en el SIG de realidad virtual online expresamente creado para esta investigación), donde los distintos servicios de mapas web se basan en la petición al servidor de unas teselas concretas, en base a la localización y el nivel de zoom que el usuario esté consultando.

Debido a lo anteriormente expuesto, se ha optado por el procesamiento de las imágenes ráster procedentes de ortofotografía aérea en forma de teselas de 256x256 píxeles, ya que éstas son las dimensiones más habituales utilizadas en cartografía y se adaptan perfectamente al trabajo realizado con las redes

generativas antagónicas, tanto para su entrenamiento como para su posterior uso en el procesamiento de imágenes.

En algunos de los experimentos realizados, se ha comprobado que los resultados pueden mejorarse, en cierta medida, utilizando tamaños de teselas mayores; pero debido a que la mejoría no resulta demasiado significativa y que, prácticamente, todos los sistemas de teselas en la cartografía digital se basan en teselas de 256x256 píxeles, se ha considerado que la mejor compatibilidad de las teselas de estas dimensiones supone una mayor ventaja que la sutil mejora obtenida en los resultados mediante el uso de teselas mayores. Esto es aplicable, principalmente, al trabajo con datos masivos de forma automática. No obstante, para el procesamiento de imágenes de pequeñas dimensiones para el estudio de un área concreta es factible la aplicación directa de la inteligencia artificial sobre la imagen sin teselar. Las dimensiones de la imagen que será capaz de procesar dependen de las capacidades de la máquina que la procese, no debiendo superar nunca, al menos con las máquinas existentes en la actualidad, las dimensiones de algunos miles de píxeles de alto por otros tantos de ancho.

2.6.1. Introducción de datos en Sistema de Información Geográfica y generación de teselas

Asumiendo que ya se han seleccionado los datos que van a procesarse, y se dispone de ellos, el primer paso consiste en cargarlos en el software SIG que vaya a utilizarse. Un ejemplo de obtención de datos y su posterior carga en QGIS se encuentra disponible en el segundo epígrafe del capítulo dos de este trabajo.

Para este ejemplo se ha optado por elegir datos correspondientes a la hoja del MTN50 del Vuelo Americano Serie B, correspondiente al área de la ciudad de Madrid, es decir la hoja MTN50 0559.

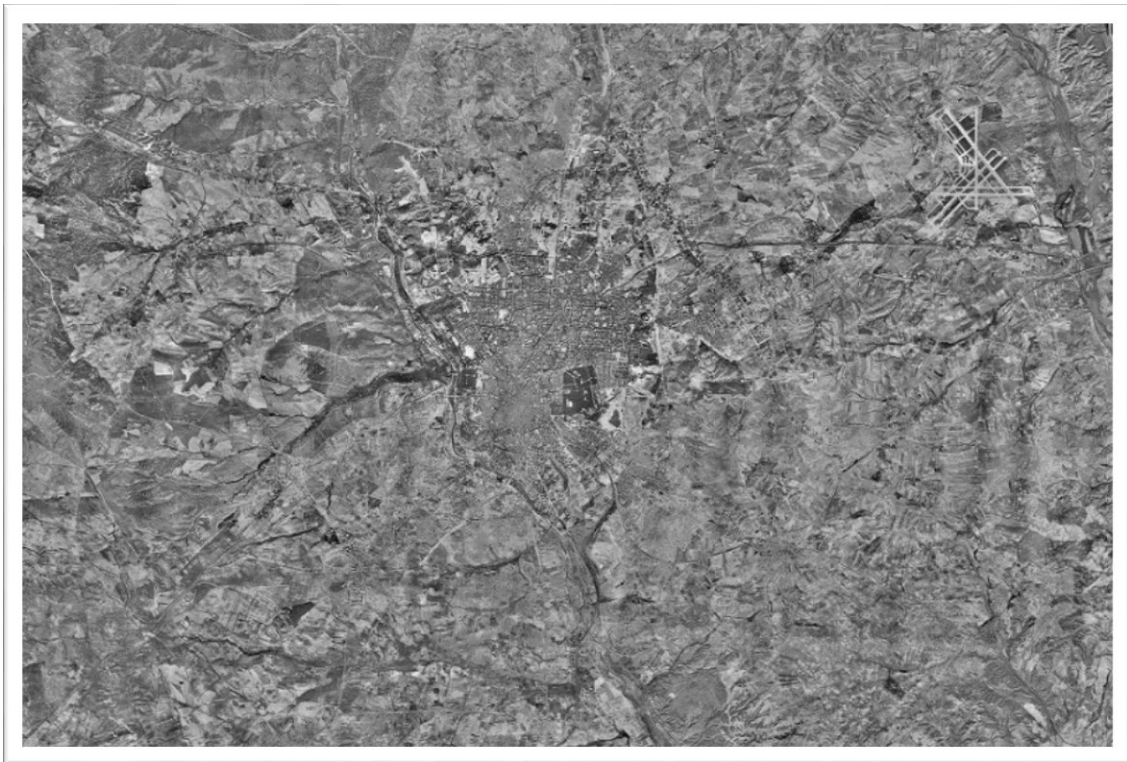


Figura 2.65. Hoja MTN50 0559 del Vuelo Americano Serie B
Fuente: elaboración propia, a partir de datos del IGN

Una vez descargados los datos y cargados éstos en el SIG, debe ajustarse el SRC, en este caso debido a la localización de la hoja se utiliza el EPSG: 25830, correspondiente con ETRS89 en el huso 30, que se corresponde con la localización de la hoja MTN50 0559.

Desde QGIS, podemos utilizar la función GdalAteselas para generar teselas a partir de nuestros datos.

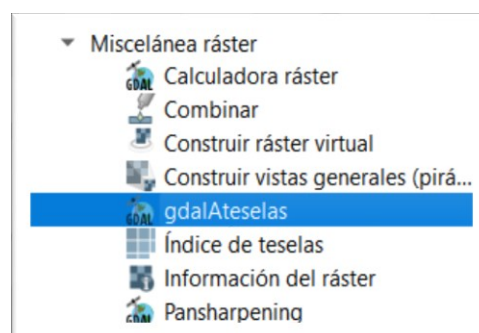


Figura 2.66. Imagen de la selección de herramienta ráster gdalAteselas incluida en QGIS
Fuente: elaboración propia

Como se ha mencionado anteriormente, las teselas estándar, de 256x256 píxeles, se adaptan bien a los modelos elaborados.

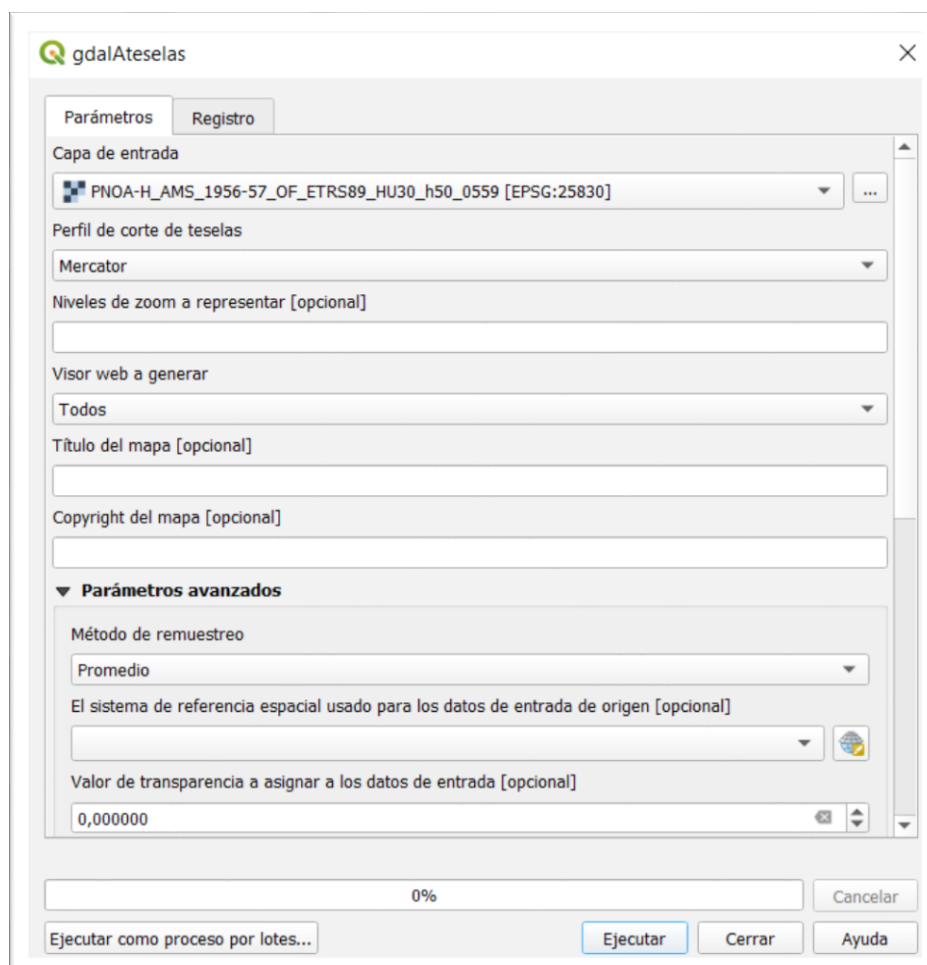


Figura 2.67. Herramienta gdalAteselas incluida en QGIS
Fuente: elaboración propia

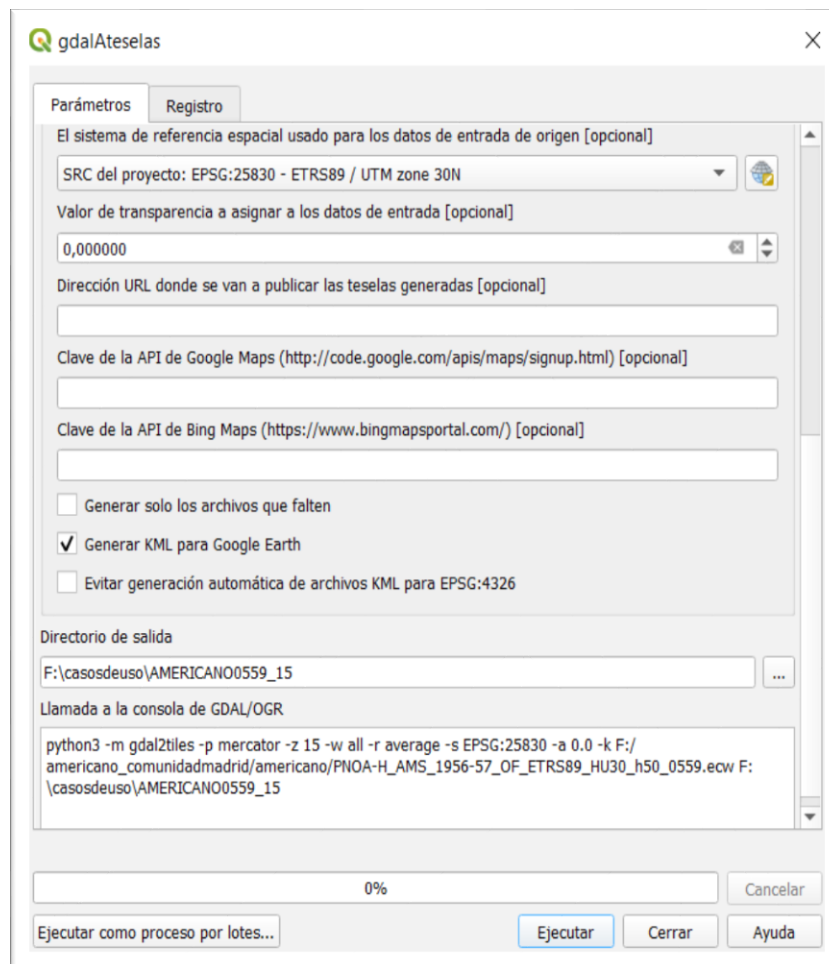


Figura 2.68. Configuración de gdalAteselas para la creación de teselas con nivel de zoom de 15 (escala aproximada 1:15.000) a partir de la hoja MTN50 0559 del Vuelo Americano Serie B
Fuente: elaboración propia

Al utilizar GDAL, las teselas generadas están en formato .png; sin embargo, para el procesamiento masivo de los datos es conveniente un formato más ligero, como .jpg. Los modelos desarrollados son capaces de procesar las imágenes tanto en png como en jpg, pero el envío de datos al servidor donde están alojados los modelos que se han creado y el procesamiento de las imágenes resulta más rápido utilizando jpg.

Para la georreferenciación de cada una de las teselas, se puede activar la opción de creación de KML, dentro de la función GdalAteselas, situada en el desplegable de parámetros avanzados.

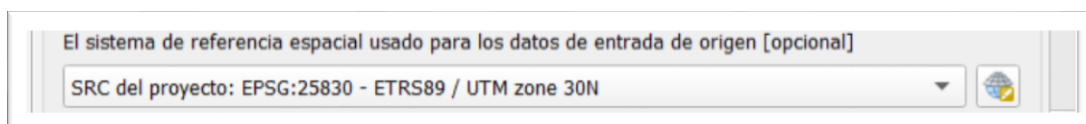


Figura 2.69. Definición del sistema de referencia espacial en la generación de teselas
Fuente: elaboración propia

En el desplegable de parámetros avanzados también puede definirse el sistema de referencia de los datos de entrada, en nuestro caso 25830 (ETRS89 / UTM zona 30N - Proyectado), tal y como se muestra en la imagen superior.

La función `GdalAteselas` genera de forma predeterminada teselas en los niveles de zoom del 10 al 16, es decir de una escala aproximada de 1:500.000 a una de 1:8.000; esta opción podría valer, ya que los modelos desarrollados, a pesar de haber sido entrenados con niveles de zoom concretos, tienen un margen de tolerancia a variaciones de 2 o 3 niveles de zoom. De todas formas, puede definirse el nivel de zoom de forma manual, en este caso escogeremos el nivel de zoom 15, o lo que es lo mismo una escala aproximada de 1:15.000, para sólo generar las teselas correspondientes a ese nivel y ahorrar por lo tanto tiempo de procesamiento y memoria.

La generación de estos archivos puede realizarse directamente utilizando la función `Gdal2tiles` desde Python, tal y como se muestra en el siguiente comando:

```
gdal2tiles -p mercator -z 15 -w all -r average -s EPSG: 25830 -a -1.0 -k
F:/americano_comunidadmadrid/americano/PNOA-H_AMS_1956-
57_OF_ETRS89_HU30_h50_0559.ecw F:\casosdeuso\AMERICANO0559_15
```

El resultado de la generación de teselas son archivos PNG y KML, ordenados por carpetas en función de sus coordenadas y de su nivel de zoom, tal y como se muestra a continuación.

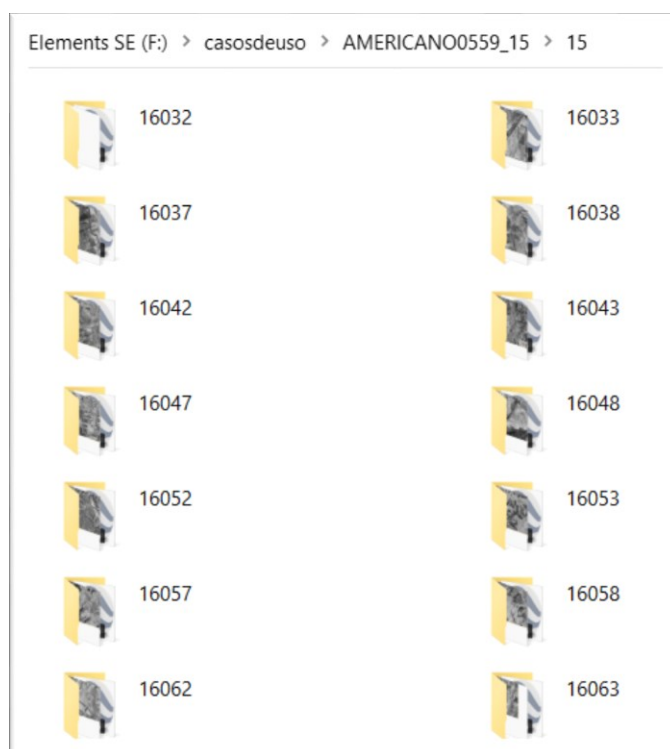


Figura 2.70. Carpetas de teselas generadas
Fuente: elaboración propia

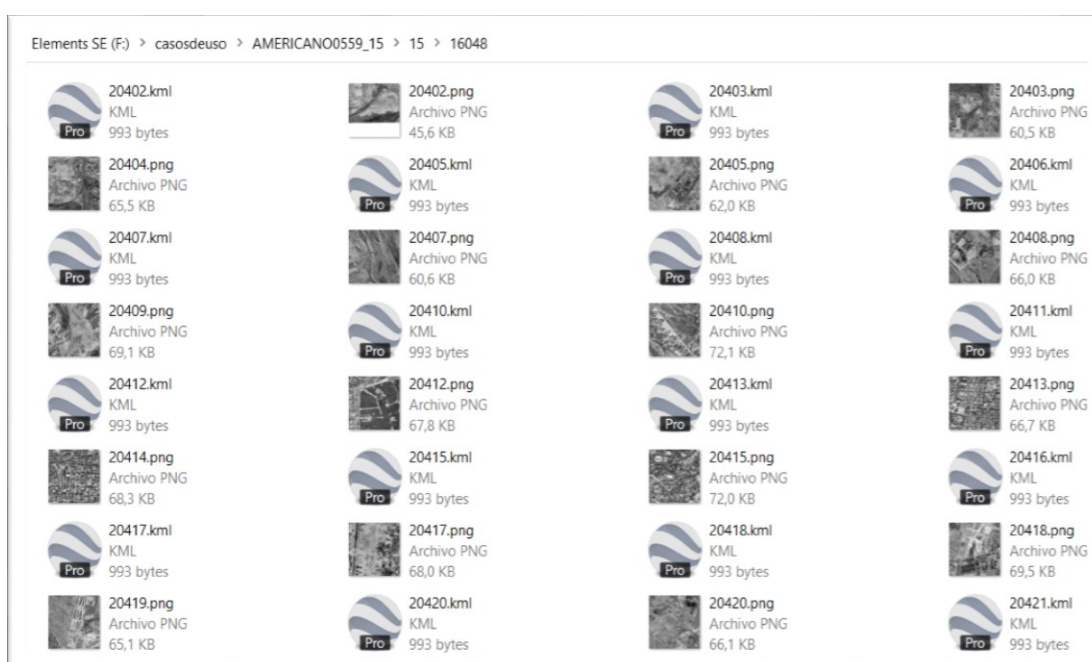


Figura 2.71. Archivos de imágenes y KML generados
Fuente: elaboración propia

Para poder agruparlos en una única carpeta y que no se repitan los nombres de los archivos, así como para cambiar de formato las imágenes a jpg y corregir las referencias a los archivos de imagen incluidas dentro de los archivos KML, se ha desarrollado un script de Python que agrupa todos los archivos en una única carpeta y los renombra incluyendo en su nombre la carpeta original que los contenía.

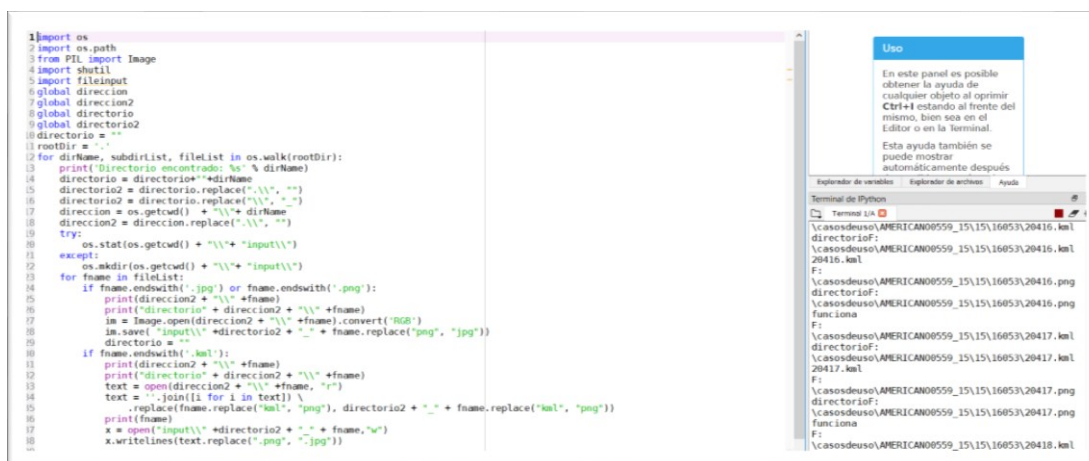


Figura 2.72. Script de Python para transformar los archivos png a jpg y corregir los KML
Fuente: elaboración propia

Es importante no olvidar corregir el contenido de los archivos KML, ya que, si no se corrigen, las referencias a los nuevos nombres y extensiones de los ficheros de imagen no serán de utilidad.

Todas estas operaciones pueden realizarse de forma automática y masiva con un sencillo script de Python como el que se muestra en la imagen superior.

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>15/16032/20405.kml</name>
    <description></description>
    <Style>
      <ListStyle id="hideChildren">
        <listItemType>checkHideChildren</listIter
      </ListStyle>
    </Style>
    <Region>
      <LatLonAltBox>
        <north>40.36328834091583</north>
        <south>40.35491675079060</south>
        <east>-3.85620117187500</east>
        <west>-3.86718750000000</west>
      </LatLonAltBox>
      <Lod>
        <minLodPixels>128</minLodPixels>
        <maxLodPixels>-1</maxLodPixels>
      </Lod>
    </Region>
    <GroundOverlay>
      <drawOrder>30</drawOrder>
      <Icon>
        <href>20405.png</href>
      </Icon>
      <LatLonBox>
        <north>40.36328834091583</north>
        <south>40.35491675079060</south>
        <east>-3.85620117187500</east>
        <west>-3.86718750000000</west>
      </LatLonBox>
    </GroundOverlay>
  </Document>
</kml>
```

Figura 2.73. Código de uno de los archivos KML generado por QGIS
Fuente: elaboración propia


```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>15/16032/20405.kml</name>
    <description></description>
    <Style>
      <ListStyle id="hideChildren">
        <listItemType>checkHideChildren</listItemType>
      </ListStyle>
    </Style>
    <Region>
      <LatLonAltBox>
        <north>40.36328834091583</north>
        <south>40.35491675079060</south>
        <east>-3.85620117187500</east>
        <west>-3.86718750000000</west>
      </LatLonAltBox>
      <Lod>
        <minLodPixels>128</minLodPixels>
        <maxLodPixels>-1</maxLodPixels>
      </Lod>
    </Region>
    <GroundOverlay>
      <drawOrder>30</drawOrder>
      <Icon>
        <href>._16032_20405.jpg</href>
      </Icon>
      <LatLonBox>
        <north>40.36328834091583</north>
        <south>40.35491675079060</south>
        <east>-3.85620117187500</east>
        <west>-3.86718750000000</west>
      </LatLonBox>
    </GroundOverlay>
  </Document>
</kml>
```

Figura 2.74. Código KML modificado por el script de elaboración propia
Fuente: elaboración propia

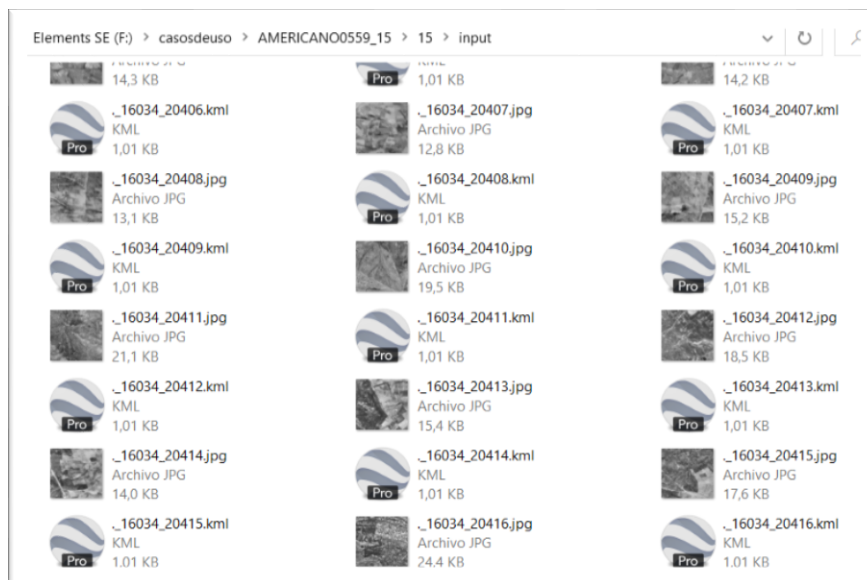


Figura 2.75. Archivos de imagen y KML modificados para poder reunirlos en una misma carpeta

Fuente: elaboración propia, a partir de datos del IGN

Una vez tenemos disponibles todos los archivos en una única ubicación, pueden comprimirse las imágenes en un archivo zip, para que su transferencia al servidor o servidores donde están alojados los modelos creados sea más rápida.

Si bien para el entrenamiento de las redes es necesaria una gran capacidad de computación, para el procesamiento de las imágenes, una vez se tiene entrenada la inteligencia artificial, el coste computacional es mucho menor, siendo posible realizar esta tarea incluso en un equipo personal. En cualquier caso, a mayor capacidad de la máquina o máquinas que se utilicen, menor será el tiempo necesario para el procesamiento de las teselas. La elección de la máquina a utilizar dependerá principalmente del tiempo disponible para realizar los cálculos y del acceso a servidores suficientemente potentes, todo ello influido, lógicamente, por el volumen de datos que se quiera procesar. Para este ejemplo se ha optado por la utilización de las máquinas de Google Colaboratory, las cuales ya se usaron para el proceso de entrenamiento y cuyas características vienen descritas en este capítulo.


```
!unzip '/content/drive/My Drive/0559americano15.zip' -d '/content/drive/My Drive/0559_15/'
inflating: /content/drive/My Drive/0559_15/._16050_20419.jpg
inflating: /content/drive/My Drive/0559_15/._16050_20420.jpg
inflating: /content/drive/My Drive/0559_15/._16050_20421.jpg
inflating: /content/drive/My Drive/0559_15/._16050_20422.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20402.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20403.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20404.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20405.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20406.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20407.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20408.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20409.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20410.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20411.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20412.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20413.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20414.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20415.jpg
inflating: /content/drive/My Drive/0559_15/._16051_20416.jpg
```

Figura 2.76. Descompresión de teselas en el servidor de Google Drive conectado a Google Colaboratory
Fuente: elaboración propia

La aplicación del modelo, una vez entrenado, resulta simple; únicamente hay que cargar el modelo generado que se quiera aplicar y alimentarlo con las teselas creadas en los pasos anteriores. Esto puede hacerse de forma sencilla con un pequeño script de Python, tal y como se muestra a continuación.

```
def generate_images(model, test_input, tar, nombres, numero):
    prediction = model(test_input, training=False)
    nombre = nombres[numero]
    tf.keras.preprocessing.image.save_img(nombre, prediction[0])
```

Figura 2.77. Definición de la función para la generación masiva de imágenes
Fuente: elaboración propia

```
for example_input, example_target in test_dataset:
    generate_images(generator, example_input, example_target, nombres, numero)
    numero = numero + 1
```

Figura 2.78. Llamada a la función de generación masiva de imágenes
Fuente: elaboración propia

Como puede apreciarse en la imagen superior, en la función de llamada se incluyen dos imágenes como parámetros; esto es debido a que la función de generación de imágenes también se ha utilizado durante los procesos de entrenamiento para comprobar la similitud del resultado con la imagen modelo utilizada en el entrenamiento.

La función de generación de imágenes masiva depende de la aplicación del modelo, el cual debe haber sido cargado previamente. La siguiente imagen muestra el código de carga del modelo generado durante el entrenamiento.

```
model = Generator()
checkpoint_dir = '/content/drive/My Drive/DeepMapsRestaur15/training_checkpoints'
checkpoint_prefix = os.path.join(checkpoint_dir, "DeepMapsRestaur15")
checkpoint = tf.train.Checkpoint(generator_optimizer=generator_optimizer,
                                discriminator_optimizer=discriminator_optimizer,
                                generator=generator,
                                discriminator=discriminator)
checkpoint.restore(tf.train.latest_checkpoint(checkpoint_dir))
checkpoint_path = checkpoint_path = "/content/drive/My Drive/DeepMapsRestaur15/training_checkpoints/DeepMapsRestaur15"
checkpoint_dir = os.path.dirname(checkpoint_path)

cp_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path,
                                                save_weights_only=False,
                                                verbose=1)

latest = tf.train.latest_checkpoint(checkpoint_dir)
latest
model.summary()
model.save('/content/drive/My Drive/deepmaps/DeepMapsRestaur15.h5')
model.save_weights(filepath="/content/drive/My Drive/deepmaps/DeepMapsRestaur15.h5")
```

Figura 2.79. Carga del modelo generado durante el entrenamiento a partir de checkpoints y guardado del mismo y de sus pesos
Fuente: elaboración propia

En el ejemplo mostrado en la imagen superior, el modelo se carga a partir del último checkpoint del entrenamiento; sin embargo, tal y como se puede ver en las dos últimas líneas de código, este modelo también se guarda (tanto el modelo como los pesos), en formato h5, el cual resulta más cómodo de cargar a la hora de implementar una aplicación definitiva. En este caso se han utilizado los checkpoints del proceso de entrenamiento debido a la naturaleza experimental del proceso.

Una vez que son procesadas las imágenes, éstas son guardadas en la misma máquina de Google Colaboratory donde han sido procesadas, desde

donde pueden comprimirse y descargarse ejecutando un par de líneas de código Python.

```
!zip -r /content/0559americanobl5.zip /content/DeepMapsRestaur15  
files.download("/content/0559americanobl5.zip")
```

Figura 2.80. Compresión de archivos procesados en Google Colaboratory
Fuente: elaboración propia

El problema de realizar el almacenamiento, tanto de los avances del entrenamiento del modelo como de las imágenes procesadas en las máquinas de Google Colaboratory, es que las sesiones de uso que permite este sistema están limitadas a unas horas, por lo que es conveniente que el código transfiera los archivos a Google Drive a medida que se van generando para que su almacenamiento sea permanente y, por lo tanto, poder realizar el proceso de forma desatendida, sin tener que preocuparse de transferir los archivos antes de que caduque la sesión.

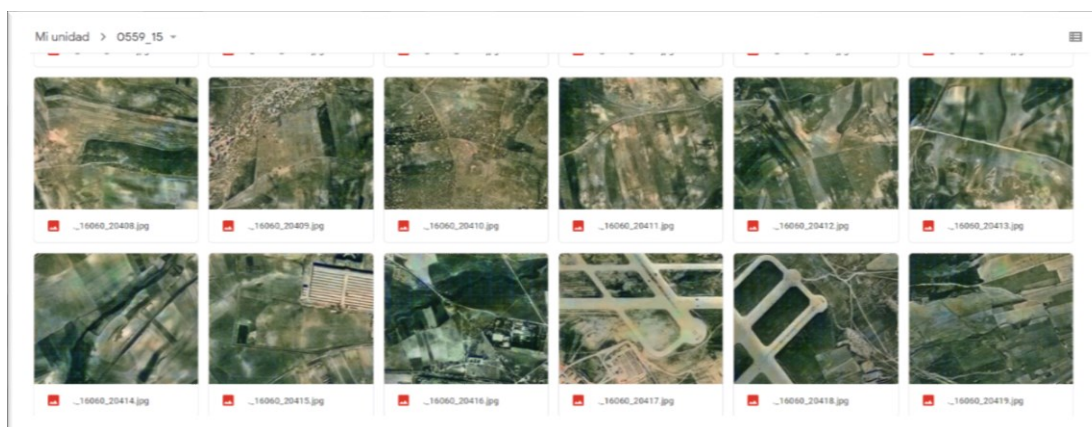


Figura 2.81. Teselas de la hoja MTN50 0559 del Vuelo Americano Serie B ya procesadas y almacenadas en Google Drive
Fuente: elaboración propia, a partir de datos del IGN

Una vez comprimidas y descargadas todas las imágenes procesadas, se combinan en una misma carpeta con los archivos KML generados en los pasos anteriores y se procede a la reconstrucción de la hoja MTN50 a partir de las teselas ya procesadas.

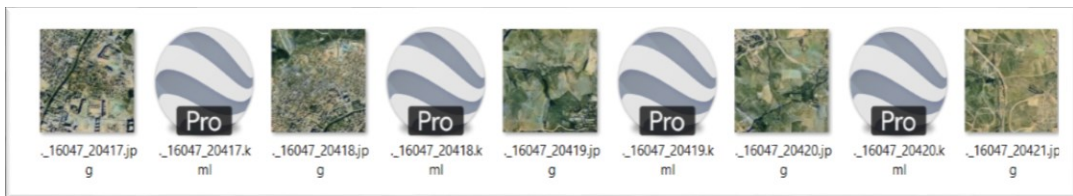


Figura 2.82. Imágenes renombradas y KML corregidos
Fuente: elaboración propia, a partir de datos del IGN

Para poder unir las teselas se recurre a QGIS, mediante la operación ráster de combinación de capas, tal y como se muestra en la siguiente imagen.

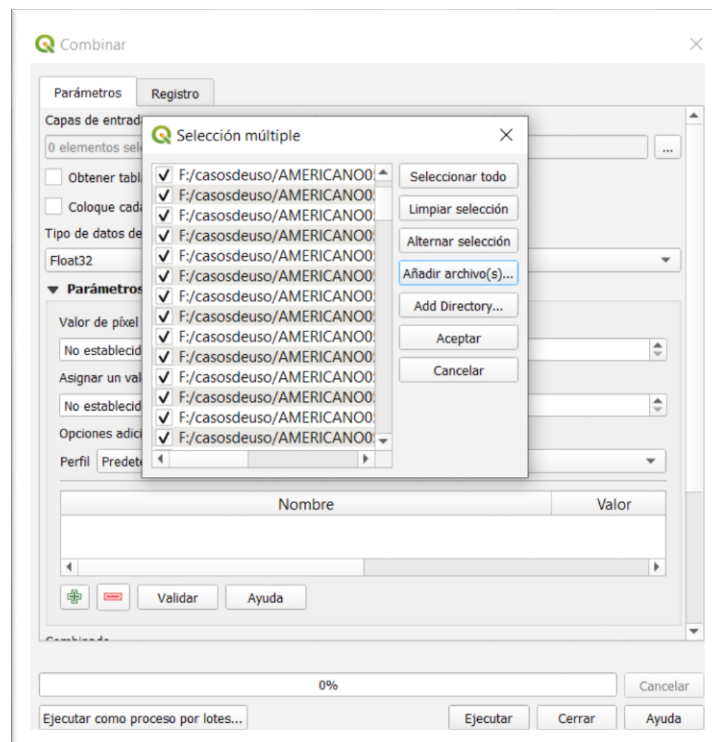


Figura 2.83. Proceso de combinación de capas ráster con QGIS
Fuente: elaboración propia

Cabe destacar que para poder restaurar una hoja del MTN50 a un nivel de zoom de 17, es decir a una escala de 1:4.000, suficiente como para que los elementos individuales sean distinguibles tanto por un humano como por una inteligencia artificial, el número de teselas de 256x256 píxeles asciende a 10.292, siendo este número mucho mayor si se trabaja con niveles de zoom más altos.

Tras las pruebas realizadas, se ha comprobado que la escala más adecuada para el procesamiento cartográfico con inteligencias artificiales es de aproximadamente 1:4.000, debido a que 17 es el nivel de zoom más bajo que ofrece resultados casi tan buenos como niveles de zoom más altos, al menos con las resoluciones de imagen disponibles en las capas del IGN. Sin embargo, el procesamiento con nivel de zoom 15 (escala aproximada de 1:15.000) ofrece resultados aceptables y permite reducir el número de teselas a 672.

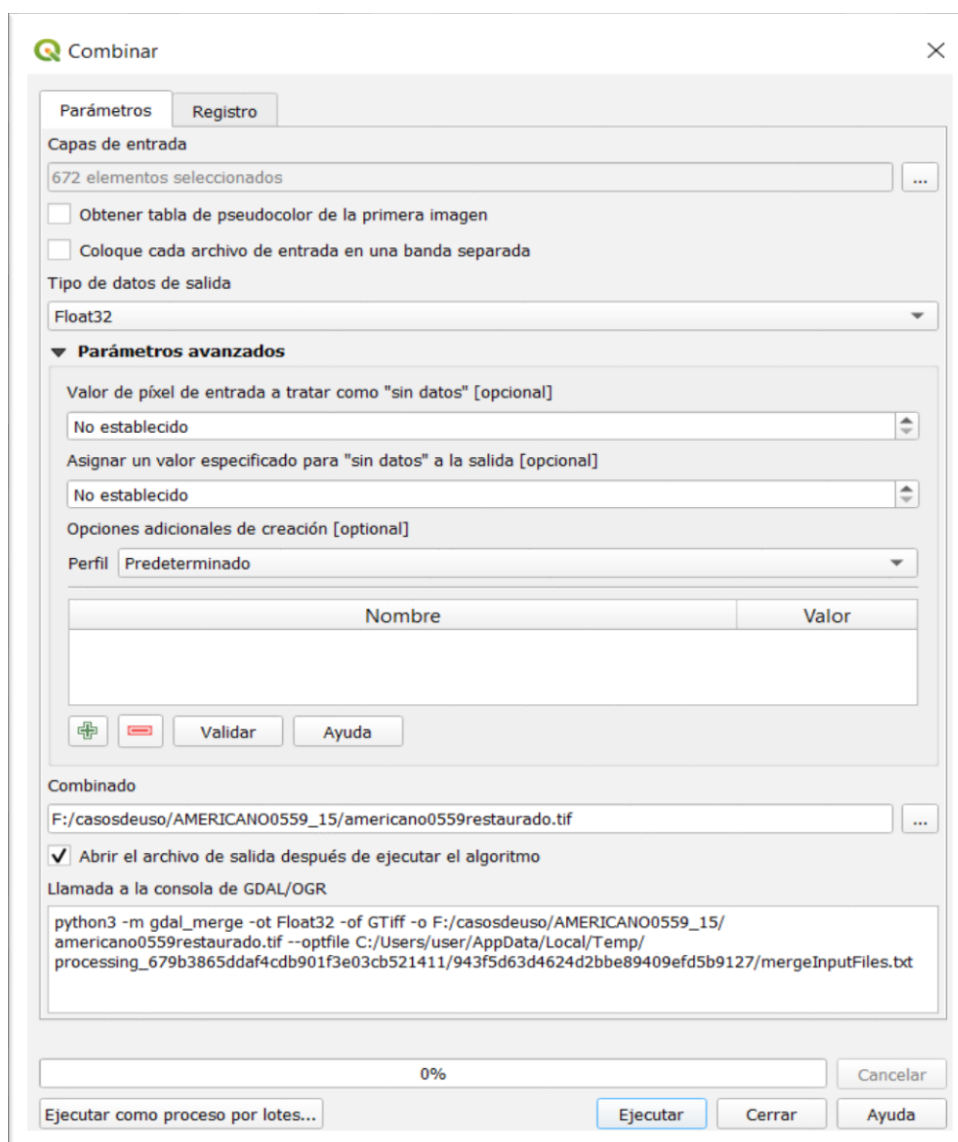


Figura 2.84. Combinación de teselas para generar la hoja MTN50 0559 del Vuelo Americano Serie B corregida
Fuente: elaboración propia

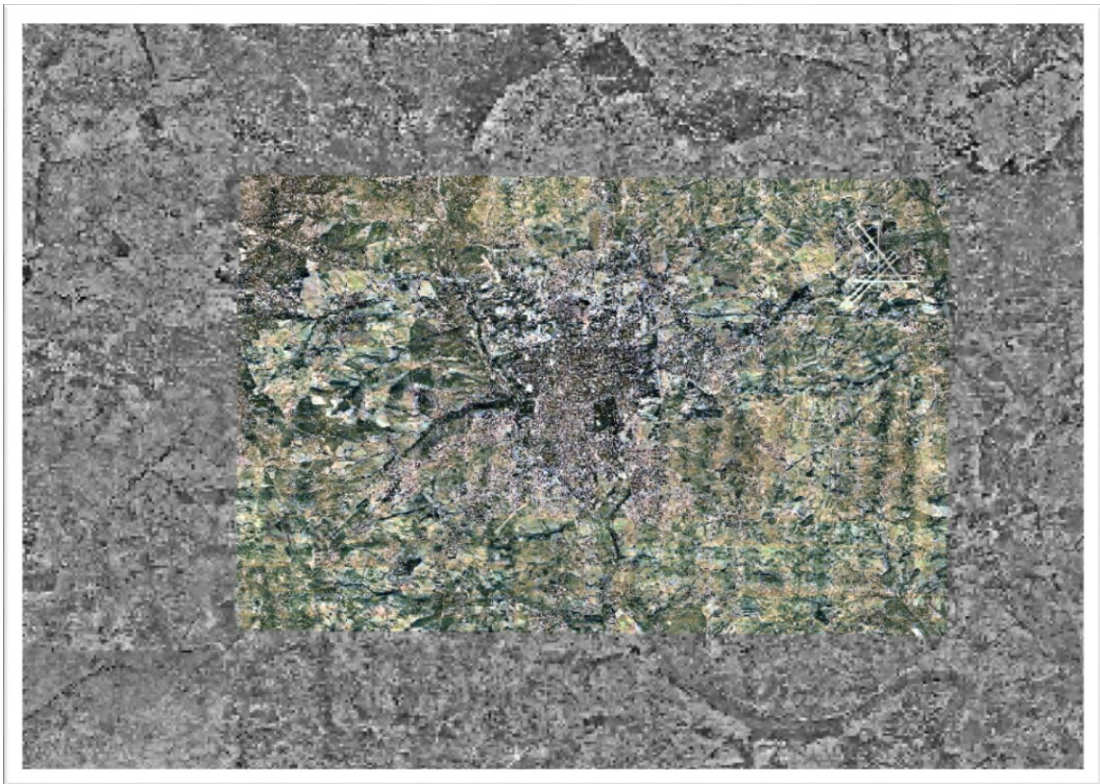


Figura 2.85. Hoja MTN50 0559 del Vuelo Americano Serie B corregida con el modelo DeepMapsRestaura15 sobre ortoimágenes del mismo vuelo sin corregir
Fuente: elaboración propia, a partir de datos del IGN

Hay que tener en cuenta que el número de teselas a procesar ha limitado las extensiones tratadas en esta tesis, ya que al tiempo necesario para teselar los datos (varias horas para los niveles de zoom entre 17 y 19, que son los más utilizados en esta investigación) hay que añadir el tiempo de tratamiento de los mismos con las inteligencias artificiales entrenadas, que si bien resulta mucho menor que el entrenamiento, supone una media de entre 2 y 5 segundos para cada tesela, a lo que hay que añadir el tiempo de unión de las teselas procesadas. A estos inconvenientes en el tiempo de procesamiento se le une la falta de servidores dedicados y debido a que se ha recurrido al uso gratuito de los servidores de Google Colaboratory, la utilización de los mismos estaba restringida a algunas horas al día, teniendo que volver a configurar el entorno en cada nueva ejecución, lo cual ha supuesto que los datos tratados se limitasen a áreas concretas.

A continuación, se muestra el detalle de algunas de las teselas de la Hoja MTN50 0559 del Vuelo Americano Serie B, corregidas con el modelo de elaboración propia (DeepMapsRestaura15), el cual, además de añadir color, corrige la luminosidad, el contraste, el enfoque y elimina errores como ruido, roturas o elementos ajenos a la ortoimagen que se hayan podido incluir durante el proceso de escaneado de las imágenes.



Figura 2.86. Ejemplo de tesela del Vuelo Americano Serie B corregida con el modelo DeepMapsRestaura15

Fuente: elaboración propia, a partir de datos del IGN



Figura 2.87. Ejemplo de tesela del Vuelo Americano Serie B corregida con el modelo DeepMapsRestaura15

Fuente: elaboración propia, a partir de datos del IGN



Figura 2.88. Ejemplo de tesela del Vuelo Americano Serie B corregida con el modelo DeepMapsRestaura15

Fuente: elaboración propia, a partir de datos del IGN

Gracias a la unión de todas las teselas generadas, ha podido componerse la hoja del MTN50 0559 del Vuelo Americano Serie B a color (Figura 2.84), y extraerse de la misma distintas áreas restauradas, como algunas de las mostradas en el caso 8 del capítulo 7 y el ejemplo expuesto a continuación, el cual incluye la Estación de Nuevos Ministerios (a la derecha), la calle Santa Engracia (a la izquierda) y la calle Ríos Rosas (Centro), todas ellas de Madrid. Las coordenadas del área representada en las figuras 2.89 y 2.90 son: 440439.24066255044, 441391.74066352204, 4476774.341688539, 4477597.301689378 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 2.89. Ortofotografía aérea original del Vuelo Americano Serie B
Fuente: elaboración propia, a partir de datos del IGN



Figura 2.90. Ortofotografía aérea generada por la inteligencia artificial DeepMapsRestaura15
Fuente: elaboración propia, a partir de datos del IGN

Algunas de las librerías utilizadas, tanto en el proceso de aplicación de los modelos como en el de entrenamiento, no vienen incluidas en las máquinas que ofrece Google Colaboratory, por lo que deben añadirse en el código Python las instrucciones por consola de comandos para su instalación en dichas máquinas.

```
!pip install google-auth==1.7.2
!pip install enum34==1.1
!pip install --upgrade --force-reinstall tensorflow-core
!pip3 install --upgrade --force-reinstall tb-nightly
!pip install -q pyyaml h5py
!pip install nvidia-smi
!nvidia-smi
```

Figura 2.91. Instalación de librerías utilizadas en el código y no instaladas en las máquinas de Google Colaboratory, junto con la llamada al servicio de monitorización de la GPU de nvidia
Fuente: elaboración propia

Tal y como ocurría en el proceso de entrenamiento de los modelos, el desarrollo del código necesario para la creación y utilización de redes neuronales se simplifica, en gran medida, gracias al uso de las librerías TensorFlow y Keras. Por otra parte, es importante recordar que, si se dispone de GPU, como en nuestro caso, para poder gestionar y monitorizar cada uno de los dispositivos Tesla, Quadro, GRID y GeForce de NVIDIA de las familias de arquitectura Fermi y superiores, es conveniente utilizar el NVIDIA System Management Interface o nvidia-smi (también NVSMI), el cual puede ser llamado por línea de comandos directamente desde el script de Python.

A continuación, puede verse un ejemplo de monitorización de la GPU Tesla T4, utilizada tanto para el entrenamiento como para el procesamiento de los datos.

```
+-----+-----+-----+
| NVIDIA-SMI 455.23.05      Driver Version: 418.67      CUDA Version: 10.1      |
+-----+-----+-----+
| GPU  Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf   Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                       |                    |                    MIG M. |
+=====+=====+=====+
|   0   Tesla T4              Off      | 00000000:00:04.0 Off |                    0 |
| N/A   37C    P8             9W / 70W |      0MiB / 15079MiB |           0%      Default |
|                                       |                    |                    ERR! |
+-----+-----+-----+

+-----+-----+-----+
| Processes:                                                         |
| GPU    GI    CI           PID   Type   Process name                      GPU Memory |
|      ID    ID                                   |             Usage |
+=====+=====+=====+
| No running processes found                                         |
+-----+-----+-----+
```

Figura 2.92. Monitorización de la GPU mediante el proceso nvidia-smi
Fuente: elaboración propia

3. El Color

En este capítulo se analizan las técnicas, existentes en la actualidad, que pueden resultar más efectivas en el proceso de coloración automatizada de ortoimágenes aéreas en escala de grises, haciendo mayor hincapié en aquellas que han devuelto mejores resultados en las pruebas realizadas y que, por lo tanto, podrían ser una opción para llevar a cabo este proceso. Sin embargo, los resultados de todas ellas muestran la necesidad de adaptar las tecnologías más vanguardistas y que han dado mejores resultados a las labores cartográficas; por ello se ha optado por la creación de inteligencias artificiales capaces de realizar, de forma automatizada, varias tareas cartográficas, como la coloración de imágenes en escala de grises para la restauración de ortoimágenes antiguas.



Figura 3.1. Restauración de ortofotografía aérea de la Puerta de Alcalá (Madrid) del Vuelo Interministerial de 1973-1986 (izquierda) mediante el Modelo DeepMapsColor17
Fuente: elaboración propia, a partir de datos del IGN

Los detalles de la elaboración de la arquitectura y otros aspectos comunes de las inteligencias elaboradas en esta investigación vienen recogidos en el capítulo 2. Al final del presente capítulo pueden encontrarse a su vez algunos detalles del entrenamiento, así como los datos con los que han sido entrenados, el número de iteraciones de su entrenamiento, etc., de los modelos que han devuelto resultados más satisfactorios en las labores de coloración automatizada de imágenes, así como algunas conclusiones que pueden extraerse de los mismos.

3.1. Adaptación de técnicas y modelos preexistentes para su aplicación en la coloración de ortoimágenes aéreas

Dada una ortofotografía aérea en escala de grises como entrada, este capítulo afronta el problema de generar una versión plausible en color de dicha imagen. Esta es una instancia específica de un problema de traducción de imagen a imagen, y tiene algunas propiedades que lo hacen más fácil que el problema general. Específicamente, la estructura de una imagen es proporcionada principalmente por la entrada en escala de grises, y los humanos son menos sensibles a los componentes de color de una imagen. Sin embargo, existen desafíos importantes para este problema. Primero, el problema está claramente poco restringido, por lo que los enfoques anteriores se han basado en una interacción significativa del usuario o han resultado en coloraciones desaturadas. En este documento se propone un enfoque totalmente automático para producir coloraciones más realistas.

En cuanto a la metodología a seguir para resolver la problemática de asignar colores a las ortofotografías aéreas en blanco y negro, existen diversas posibilidades que vamos a explorar a continuación. Estas posibilidades pueden ser más simples o complejas computacionalmente. Partiendo desde las más simples, como sería la extracción de los colores de una imagen del mismo territorio en color, y la asignación de éstos a la imagen correspondiente a ese territorio en blanco y negro, a las más complejas basadas en la utilización de redes neuronales generativas antagónicas. Dentro de la utilización de redes neuronales, también podría distinguirse entre la utilización de redes preentrenadas para asignar colores a fotografías en blanco y negro o el entrenamiento de forma específica de una red que sea capaz de asignar estos colores. Las diferencias entre estas dos últimas técnicas, utilizando redes neuronales, residen principalmente en los requisitos de computación necesarios, ya que entrenar una red neuronal exige una gran capacidad de potencia de cálculo, así como un DataSet de imágenes adecuado para su entrenamiento.

La ventaja principal que nos ofrecen las redes neuronales ya entrenadas es que su aplicación es mucho más inmediata, puesto que ya han sido entrenadas con grandes volúmenes de datos; sin embargo, el problema que presentan reside en que los datos con los que han sido entrenadas son principalmente de fotografías ordinarias en color, no de imágenes aéreas, por lo que son especialmente eficaces a la hora de detectar elementos cotidianos como personas u objetos, pero no tanto a la hora de asignar colores a las tomas aéreas. A pesar de ello, como muchas de las imágenes con las que han sido entrenadas contienen paisajes, los resultados obtenidos al aplicar este tipo de redes neuronales a la coloración de imágenes aéreas en escala de grises son bastante aceptables. Por otra parte, el entrenamiento de una red neuronal a partir de datos propios procedentes de imágenes aéreas actuales en color nos ofrece la posibilidad de que esta red reconozca los elementos típicos presentes en las imágenes aéreas; sin embargo, se le añade la dificultad de generar el set de datos necesario para su entrenamiento, así como de disponer de la capacidad de cálculo necesaria para el mismo. Por lo tanto, a continuación, vamos a explorar distintas pruebas realizadas con sus correspondientes resultados, obtenidos al aplicar las técnicas anteriormente descritas y a compararlos con los producidos por los modelos desarrollados.

3.2. Antecedentes

Los primeros métodos semiautomáticos utilizaban un enfoque basado en ejemplos; estos métodos transferían estadísticas de color desde una imagen de referencia o múltiples imágenes a la imagen en escala de grises de entrada, con técnicas como la transferencia de color, de la cual se ha incluido un ejemplo en el siguiente epígrafe. Estos métodos funcionan notablemente bien cuando la entrada y la referencia comparten contenido similar. Sin embargo, encontrar imágenes de referencia lleva mucho tiempo y puede ser un desafío para objetos poco habituales o escenas complejas, incluso cuando se utilizan métodos de recuperación semiautomáticos y, a pesar de que los datos georreferenciados suponen una ayuda, en la práctica, las ortoimágenes aéreas que han sido captadas en distintos años, muestran notables variaciones entre sí. Además, algunos algoritmos implican grandes esfuerzos manuales para definir las regiones correspondientes entre las imágenes.

Recientemente, se han propuesto métodos totalmente automáticos, como los propuestos en esta tesis. Los métodos recientes consisten en entrenar Redes Neuronales Convolucionales con colecciones de imágenes, para crear directamente su correspondencia entre imágenes en escala de grises y los colores de salida. Las redes pueden aprender a combinar señales de bajo y alto nivel para realizar la coloración, y se ha demostrado que producen resultados más realistas.

Otra de las características de las redes neuronales profundas es que tienen un desempeño destacable en extraer semántica a partir de imágenes, desde conceptos de nivel medio, como material y segmentación, hasta conocimiento de alto nivel, como objetos y categorías de la escena. Dicho con otras palabras, las redes profundas son capaces de percibir los elementos presentes en una imagen y entenderlos como conceptos a partir de los cuales pueden operar, superando los conceptos de bajo nivel como las características de píxeles

individuales con los que trabajan los algoritmos tradicionales, llegando a poder extraer conceptos de nivel medio, como las texturas presentes en la imagen o la posible segmentación de las mismas en distintas categorías (lo cual resulta especialmente útil para la generación automatizada de mapas de usos del suelo) e incluso pudiendo extraer conceptos de alto nivel, de tal forma que pone a las inteligencias artificiales profundas, al nivel de la percepción visual humana, pudiendo percibir objetos individuales (factor éste que es esencial a la hora de asignar colores plausibles a las imágenes en blanco y negro, distinguiendo cada uno de los objetos presentes en la imagen y coloreándolos en base a las características de otros objetos similares aprendidas durante el entrenamiento) y categorías de la escena, es decir pueden categorizar los conjuntos de elementos presentes en distintas imágenes para así diferenciar una escena urbana de una costera o de una de monte. Gracias a esto las inteligencias artificiales desarrolladas en este trabajo son capaces de realizar de forma automática tareas que hasta ahora sólo podían realizar los humanos, algunas de ellas requiriendo gran esfuerzo y dedicación.

Toda esta información podría beneficiar potencialmente la edición de ortoimágenes con una interpretación semántica de las mismas, es decir, cambiar el contenido visual de alto nivel con una interacción mínima del usuario y dar resultados verosímiles; un buen ejemplo de ello es la inteligencia de creación propia DeepMapsInpainting, a través de la cual pueden eliminarse elementos de una imagen y recrearla, de forma verosímil, sin ellos (en el capítulo 5, en la Figura 5.23 se incluye un ejemplo de eliminación de un edificio mediante el uso de esta inteligencia artificial).

Recientemente, las redes neuronales han mostrado resultados impresionantes para diversas tareas de procesamiento de imágenes, como la mejora de la calidad fotográfica, la cual se analiza en los capítulos 4 y 5 y para lo que se proponen también una serie de modelos propios.

3.3. Transferencia de color entre imágenes

Debido a que los datos con los que tratamos corresponden a ortoimágenes aéreas que están georreferenciadas, la forma más obvia y eficiente de dar color a una imagen en blanco y negro sería transferir el color de la imagen actual, correspondiente a una ubicación concreta, a imágenes de la misma ubicación, pero tomadas en escala de grises. De esta forma, en teoría, podrían aprovecharse los datos en color de un área determinada para transferirlos a imágenes antiguas de esa misma área, donde en su día no se pudo capturar el color o la captura de este fue deficiente.

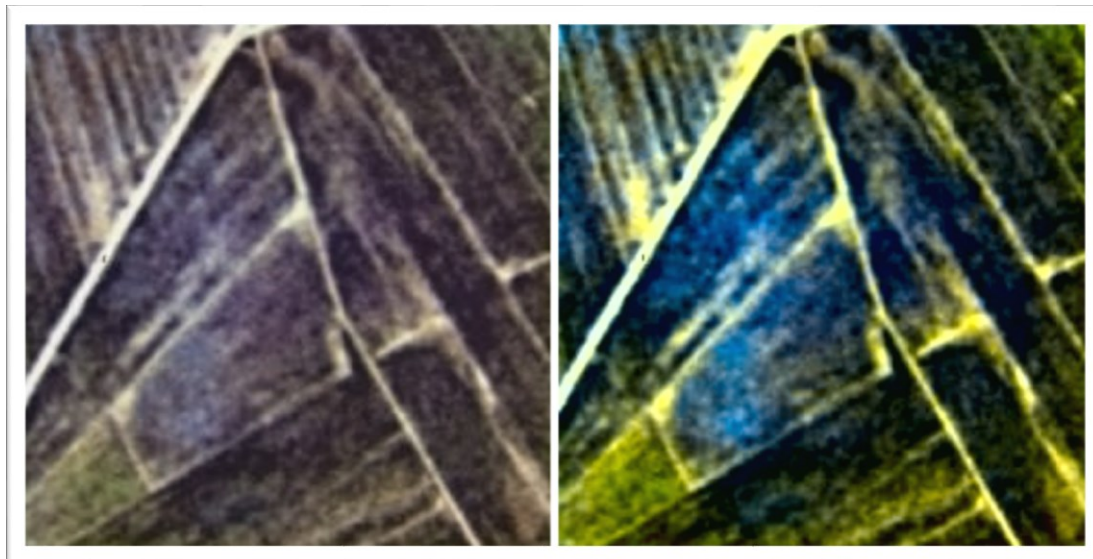


Figura 3.2. Comparativa entre la imagen original del Vuelo Americano Serie A (izquierda) y la imagen coloreada (derecha) mediante el método de transferencia de colores entre imágenes, el color ha sido tomado de la capa PNOA de Máxima Actualidad

Fuente: elaboración propia, a partir de datos del IGN

Lógicamente para que el proceso pudiese ser aprovechable, debía crearse un algoritmo que pudiera tomar dos imágenes, una fuente y un objetivo, y luego transferir el espacio de color de la imagen actual a la imagen antigua y que, además, permitiese el tratamiento de forma masiva.

A este respecto, ya se han desarrollado algunos algoritmos para la transferencia de colores entre imágenes, centrados en el ámbito de la edición fotográfica. Los algoritmos existentes se basan en histogramas; este enfoque obtiene buenos resultados en lo que se refiere a fotografías habituales, paisajes, personas, etc., pero consume demasiados recursos de computación. El uso de este algoritmo requeriría la realización de una búsqueda para todos y cada uno de los píxeles de la imagen de origen, lo que se volverá extremadamente costoso a medida que la imagen aumente de tamaño o lo haga el volumen de imágenes a tratar, como ocurre en el caso de la cartografía. Estos impedimentos de costes de procesamiento descartan este método, de modo que el siguiente planteamiento se basa en crear un algoritmo de transferencia de color que no use nada más que la desviación media y estándar de los canales de imagen, lo cual acelera enormemente el procesamiento de las imágenes y permite el tratamiento masivo de las mismas. El algoritmo utilizado está basado en el trabajo de Reinhard et al, publicado en 2001 “Color Transfer between Images”. En este artículo, Reinhard y sus colegas demuestran que al utilizar el espacio de color $L^*a^*b^*$ y la desviación media y estándar de cada canal L^* , a^* y b^* respectivamente, el color se puede transferir entre dos imágenes.

El funcionamiento del algoritmo es el siguiente: Se ingresan dos imágenes, una de origen, de la cual se va a extraer el color, y otra de destino, la cual va a ser coloreada con los datos de la imagen de origen. Tras esto, deben transformarse tanto la imagen de origen como la de destino al espacio de color $L^*a^*b^*$. El espacio de color $L^*a^*b^*$ hace un trabajo sustancialmente mejor, imitando como los humanos interpretan el color, que el espacio de color RGB estándar, este sistema funciona bastante bien para las imágenes habituales y los procesos fotográficos comunes. Tras esto se dividen los canales para la imagen de origen y la de destino, y se calcula la desviación estándar y la media de cada uno de los canales $L^*a^*b^*$ para las dos imágenes. Posteriormente, se resta la media de los canales $L^*a^*b^*$ de la imagen de origen a la de los canales de la imagen de destino y se escalan los canales de la imagen de destino por la relación de la desviación estándar de las imágenes de origen y de destino (esta

relación se obtiene dividiendo la desviación estándar de la imagen de destino entre la desviación estándar de la imagen de origen). Para realizar el escalado de cada uno de los canales de la imagen, se multiplica la relación entre las desviaciones estándar anteriormente obtenidas por cada uno de los canales de la imagen destino.

Por último, se agregan las medias de los canales $L^*a^*b^*$ de la imagen fuente y se recortan los valores que estén fuera del rango $[0, 255]$. Por último, se vuelven a fusionar los canales y se convierte de nuevo al espacio de color RGB desde el espacio $L^*a^*b^*$.

La implementación de este algoritmo resulta sencilla, ya que gran parte de los procesos vienen implementados en las librerías de Python Numpy y OpenCV.

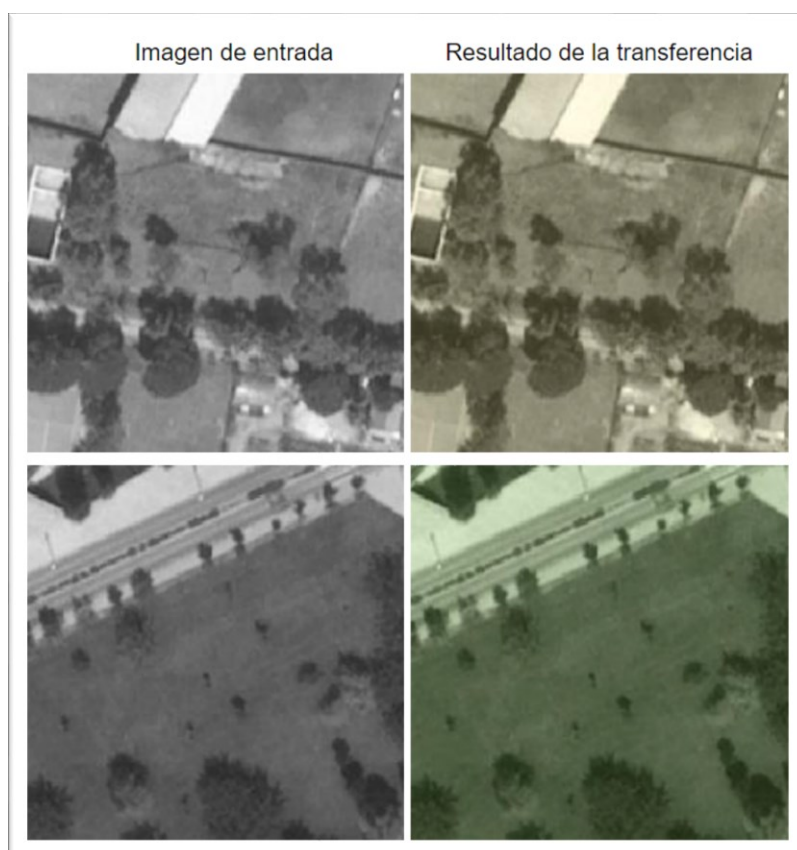


Figura 3.3. Resultados obtenidos al transferir los datos de color de imágenes del PNOA a esas mismas imágenes en escala de grises
Fuente: elaboración propia, a partir de datos del IGN

Como puede comprobarse en los resultados, los colores y la tonalidad general asignados se corresponden con las imágenes actuales y se mantienen los elementos presentes en las imágenes en blanco y negro originales. Sin embargo, la asignación de estos colores no se corresponde exactamente a los distintos elementos presentes en la imagen; esto es debido a que el algoritmo fue diseñado para transferir colores entre imágenes ya coloreadas y no para dar color a imágenes en blanco y negro, por lo que las únicas referencias para la asignación de color son las que corresponden al canal L. Debido a esto, el algoritmo de transferencia de color, en principio, queda descartado para el propósito que nos ocupa; sin embargo, puede resultar interesante su aplicación para la mejora de resultados obtenidos con técnicas de coloración basadas en redes neuronales.

3.4. Colorización de imágenes mediante Deep Learning

La técnica aplicada en este epígrafe está recogida en el paper ECCV 2016 de Zhang et al., Colourful Image Colorization.

Hasta la aplicación de las redes neuronales, los enfoques para la coloración de imágenes en blanco y negro se basaban en la anotación humana realizada de forma manual y, a menudo, esta metodología producía resultados desaturados, que no eran "creíbles" como coloraciones verdaderas.

Zhang y su equipo decidieron atacar el problema de la coloración de la imagen mediante el uso de redes neuronales convolucionales para predecir cómo se vería una imagen de escala de grises cuando se colorea.

Para entrenar a la red se comenzó con el conjunto de datos ImageNet y se convirtieron todas las imágenes del espacio de color RGB a CIE $L^*a^*b^*$ (CIELAB), proceso que también se ha llevado a cabo para el entrenamiento de los modelos de coloración creados en esta investigación.

Al igual que el espacio de color RGB, el espacio de color CIE $L^*a^*b^*$ (CIELAB) tiene tres canales. Pero la diferencia del espacio del color RGB con el espacio de color $L^*a^*b^*$ es que este último codifica la información de color de manera diferente; en este espacio de color a es un canal que codifica rojo-verde y el canal b codifica azul-amarillo. Además, el canal L codifica sólo intensidad de luminosidad; este hecho es fundamental, ya que, mediante el uso de este canal, se han generado las imágenes en escala de grises, necesarias para el entrenamiento de varios de los modelos.

Sería demasiado largo explicar los pormenores del espacio $L^*a^*b^*$, pero lo esencial es que $L^*a^*b^*$ hace un mejor trabajo representando la forma en la que los humanos vemos el color.



Figura 3.4. Imagen original en escala de grises e imagen coloreada mediante la red propuesta en Colourful Image Colorization por Zhang et al.
Fuente: elaboración propia, a partir de datos del IGN

Al igual que en los modelos generados en este estudio, en el entrenamiento de esta red se aprovecha que el canal L codifica sólo la intensidad, y se utiliza este canal como la entrada de escala de grises a la red. A partir de ahí, la red aprende a predecir los canales a^* y b^* . Dado el canal L de entrada y los canales a^*b^* predichos, se forma la imagen de salida final.

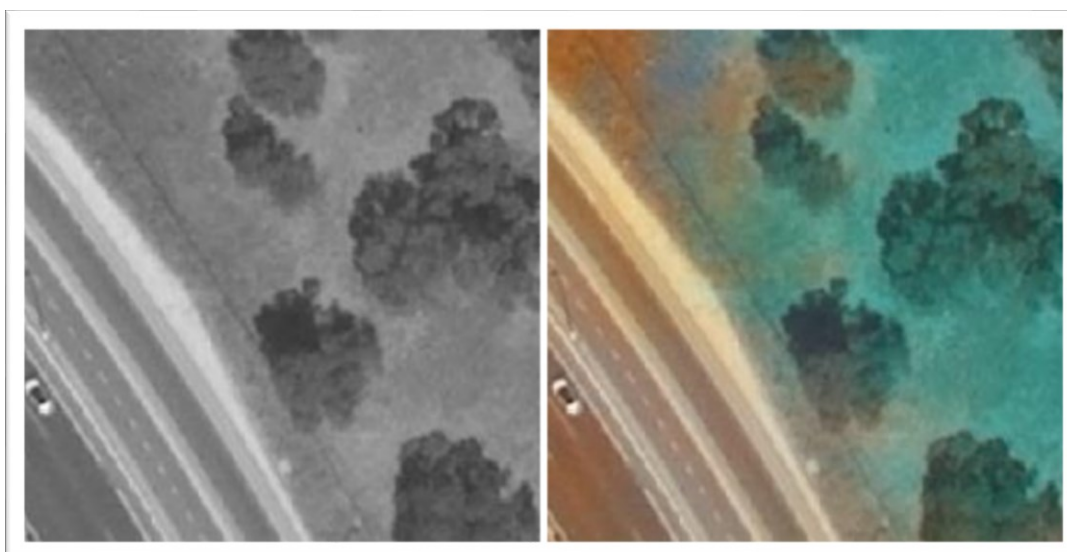


Figura 3.5. Imagen original en escala de grises e imagen coloreada mediante la red propuesta en Colourful Image Colorization por Zhang et al.
Fuente: elaboración propia, a partir de datos del IGN

Para obtener más detalles sobre el algoritmo de colorización de imágenes y el modelo de aprendizaje profundo, en el que se basa la prueba recogida en este epígrafe, puede revisarse el trabajo de Zhang et al., cuya referencia viene incluida en la bibliografía. Cabe mencionar, además, la importancia de sistemas similares, como los presentados por Larsson et al. e Iizuka et al., los cuales también aprovechan el Big Data y las Redes Neuronales Convolucionales, incluidos en la bibliografía.

Estos métodos y los creados en esta investigación para el trabajo cartográfico, difieren en las arquitecturas de sus Redes Neuronales Convolucionales y sus funciones de pérdida. La arquitectura de Larsson et al. utiliza hipercolumnas en una red VGG; Iizuka et al. utiliza una arquitectura de dos flujos en la que fusionan características globales y locales. En esta prueba se utiliza una red de un solo flujo, de estilo VGG, con profundidad adicional y convoluciones dilatadas; sin embargo, en las redes neuronales creadas se utiliza una red U-Net en el generador y una red PatchGAN en el discriminador utilizado durante el entrenamiento. Evidentemente, otra diferencia sustancial son los datos con los que los distintos modelos han sido entrenados; mientras que en esta prueba y en los trabajos de Larsson et al. los modelos son entrenados con los conjuntos de imágenes correspondientes al dataset ImageNet, Iizuka et al. entrenan a su modelo con imágenes procedentes de Google Street View, mientras que los modelos aquí desarrollados han sido entrenados con ortoimágenes aéreas obtenidas en su mayoría a partir de las capas ofrecidas por el IGN. Existen más detalles del entrenamiento de los modelos generados para los procesos de restauración y coloración automatizada aplicados a la cartografía al final de este capítulo.



Figura 3.6. Imagen original en escala de grises (Izquierda) e imagen coloreada mediante la red propuesta en Colourful Image Colorization por Zhang et al. (Derecha)
Fuente: elaboración propia, a partir de datos del IGN

Cómo puede comprobarse en los resultados, se produce una asignación más exacta de colores a los distintos elementos que en la prueba del epígrafe anterior; sin embargo, al observar en detalle las imágenes pueden apreciarse una serie de anomalías cromáticas.

A continuación, analizaremos la posibilidad de mejorar los resultados de la aplicación de estas redes neuronales preentrenadas mediante la transferencia de estadísticas de color correspondientes a una imagen actual del mismo territorio.

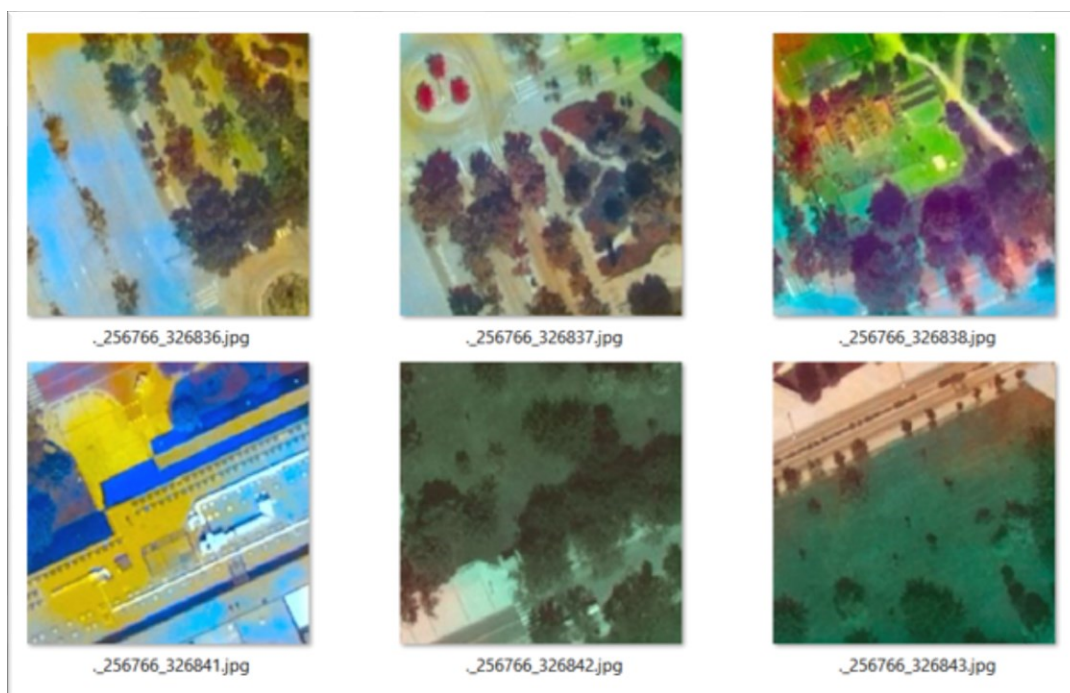


Figura 3.7. Resultados de la combinación de la red neuronal de colorización propuesta por Zhang et al. y la técnica de transferencia de colores
Fuente: elaboración propia, a partir de datos del IGN

Como puede comprobarse en los resultados, la combinación de estas dos técnicas aumenta la coloración, pero da unos resultados poco realistas en un intento de añadir colores a la imagen ya coloreada erróneamente en un principio por redes neuronales que no han sido creadas específicamente para el trabajo con ortoimágenes.

3.5. La transferencia de estilo

El modelo de red escogido en esta prueba para la coloración de las ortoimágenes aéreas no ha sido diseñado específicamente para la coloración de imágenes, sino para cuestiones artísticas mediante la transferencia de estilos. En este documento se analiza la posibilidad de tomar el color y la resolución de una imagen como estilo y por lo tanto poder aprovechar la potencia de esta tecnología. Para comprobar si la hipótesis planteada es válida se ha procedido a la realización de diferentes pruebas, cuyos resultados se muestran al final de este epígrafe. A grandes rasgos, el funcionamiento de este modelo se basa en la alimentación de éste con dos imágenes, una imagen de estilo y otra de contenido, lo cual produce activaciones en todas sus capas. De ahí extraemos dos subconjuntos de activaciones, las de estilo (naranja) y las de contenido (verde), a diferentes profundidades en la red (Figura 3.8).

Para entender estos conceptos hay que recordar que las redes neuronales profundas, como las que componen los modelos desarrollados están compuestas por distintas capas, las cuales, a su vez, están formadas por neuronas que se conectan entre sí para poder transmitirse mensajes y compartir información que será relevante para generar los distintos outputs de la capa. Normalmente, las capas que componen una red neuronal profunda se dividen en tres categorías input, hidden, output; las capas de entrada o input son las encargadas de introducir los datos de la red, las capas hidden pueden ser muy numerosas y sus funciones no resultan evidentes; por último, las capas output son las encargadas de generar los datos de salida. Dentro de estas capas hidden, u ocultas en las redes convolucionales entrenadas y alimentadas con imágenes, se ha podido comprobar que capas más o menos profundas contienen las abstracciones de estilo o de contenido, entendiendo el estilo como la configuración de colores, texturas, etc. y contenido como los elementos presentes en las imágenes. Por otra parte, el concepto de activaciones se refiere a la activación de las neuronas contenidas en las capas, que en el caso de las

redes neuronales artificiales resulta muy similar a la activación biológica de las neuronas, estando determinada, en el caso de las redes neuronales artificiales, por una función de activación, pudiendo esta ser de varios tipos, lineales o no lineales, como es el caso de las funciones de activación utilizadas en las inteligencias artificiales creadas. Por lo tanto, las activaciones de estilo se refieren a la activación de neuronas en las capas relacionadas con el estilo y las activaciones de contenido se refieren a la activación de neuronas en las capas relacionadas con el contenido.

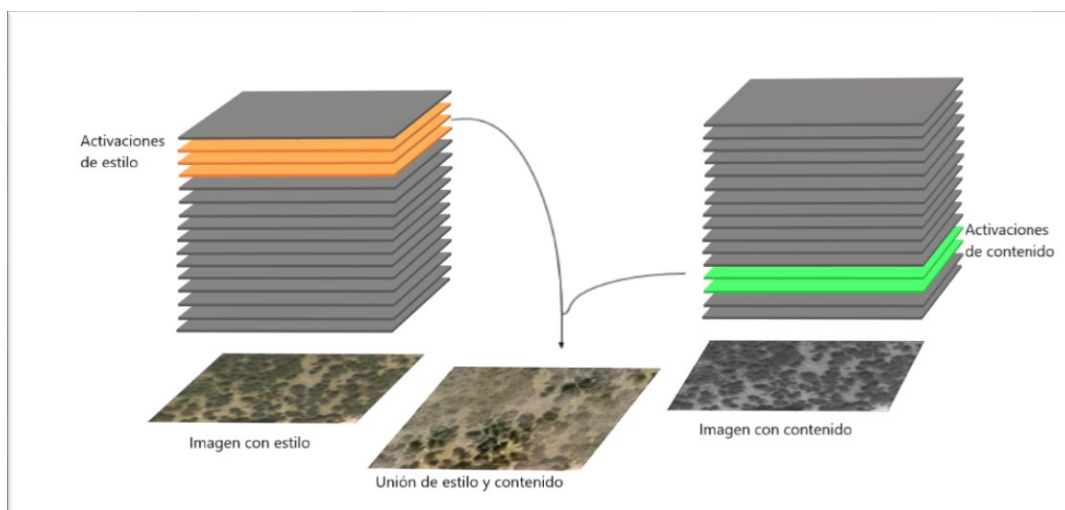


Figura 3.8. Esquema del funcionamiento de la transferencia de estilo mediante Deep Learning
Fuente: elaboración propia

Luego, se optimiza una entrada para producir activaciones similares al conjunto combinado de activaciones que fueron extraídas. La elección de las capas que mejor funcionan debe hacerse revisando los resultados obtenidos como producto de distintas combinaciones. A continuación, pueden verse dos imágenes y algunas de las combinaciones de las capas extraídas de ellas, así como una ampliación de la combinación que ha producido resultados más satisfactorios.

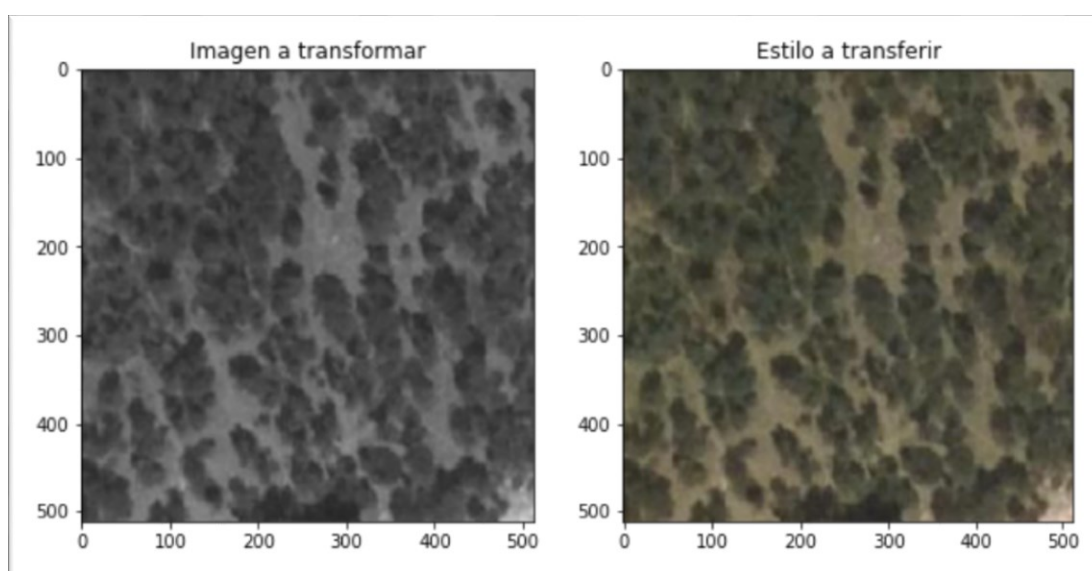


Figura 3.9. Imágenes de entrada utilizadas para la prueba de coloración mediante la transferencia de estilo con Deep Learning
Fuente: elaboración propia, a partir de datos del IGN



Figura 3.10. Resultados de colorización obtenidos mediante la transferencia de estilo, extrayendo las capas de estilo y contenido a distintas profundidades de la red neuronal
Fuente: elaboración propia, a partir de datos del IGN

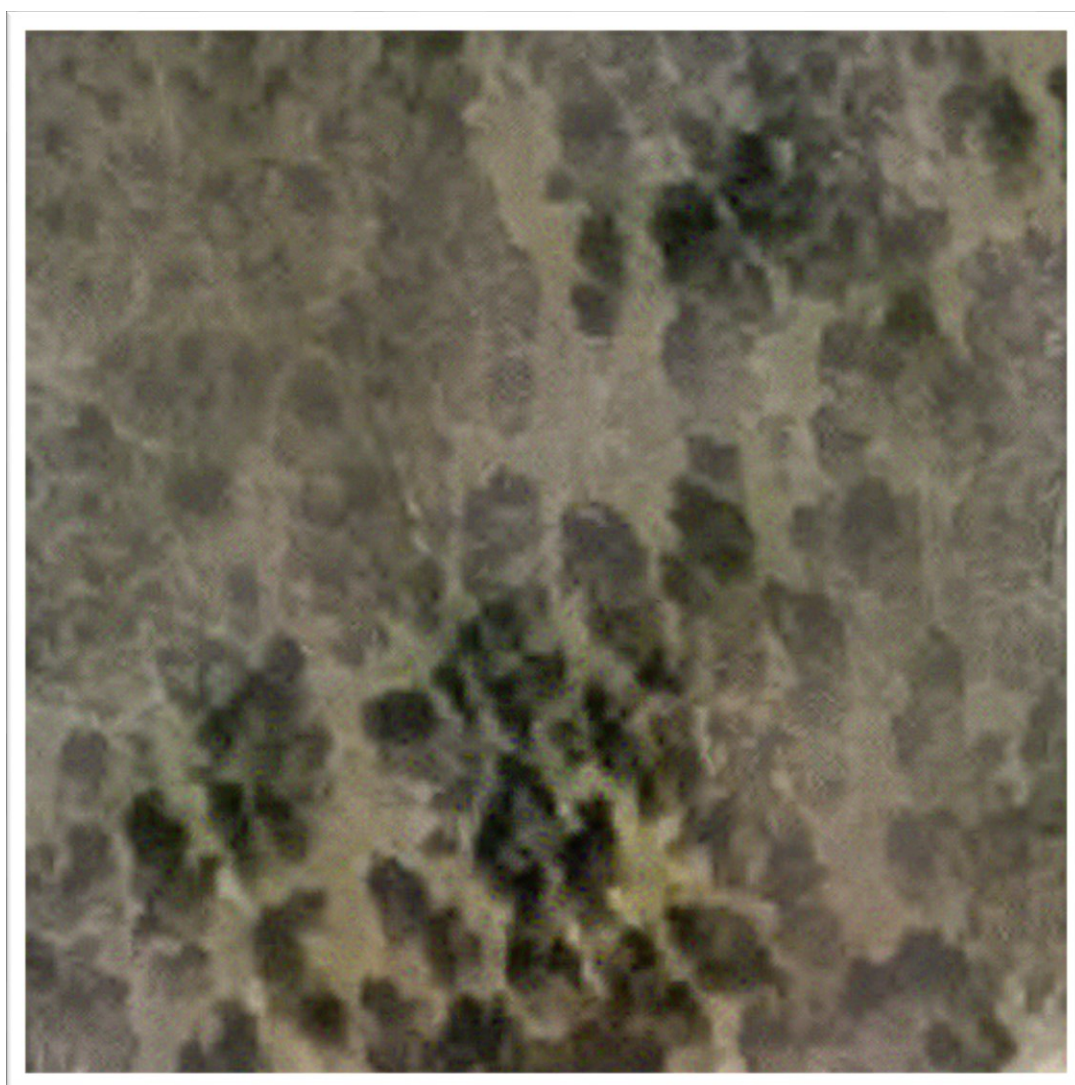


Figura 3.11. Ampliación del mejor resultado de colorización obtenido mediante aplicación de la transferencia de estilo con Deep Learning
Fuente: elaboración propia, a partir de datos del IGN

Como ya se ha comentado en la introducción, las Redes Neuronales Profundas que son más eficaces en tareas de procesamiento de imágenes son llamadas Redes Neuronales Convolucionales. Las Redes Neuronales Convolucionales consisten en capas de pequeñas unidades de cálculo que procesan la información visual de forma jerárquica. Cada capa de unidades puede entenderse como una colección de filtros de imagen, cada uno de los cuales extrae una cierta característica de la imagen de entrada. Así, la salida de una capa dada consiste en los llamados "mapas de características", que constituyen versiones filtradas de forma diferente de la imagen de entrada.

Cuando las Redes Neuronales Convolucionales se entrenan en el reconocimiento de objetos, desarrollan una representación de la imagen que hace que la información del objeto sea cada vez más explícita a lo largo de la jerarquía del proceso, transformándose en representaciones cada vez más focalizadas en el contenido real de la imagen en comparación con sus valores de píxeles detallados. Por lo tanto, se puede visualizar directamente la información contenida en cada capa sobre la imagen de entrada reconstruyendo la imagen sólo a partir de los mapas de características en los que las capas superiores de la red capturan el contenido de alto nivel en términos de objetos de la imagen de entrada, pero sin limitar los valores exactos de los píxeles de la reconstrucción. Por el contrario, las reconstrucciones de las capas inferiores permiten reproducir los valores exactos de los píxeles de la imagen original.

Por lo tanto, podemos referirnos a los mapas de características en las capas superiores de la red como la representación del contenido de alto nivel.

Para obtener una representación del estilo de una imagen de entrada, usamos un espacio de características originalmente diseñado para capturar información de la textura. Este espacio de características está construido sobre las respuestas de los filtros en cada capa de la red. Consiste en las correlaciones entre las diferentes respuestas de los filtros sobre la extensión espacial de los mapas de características. Al incluir las correlaciones de múltiples capas, obtenemos una representación estacionaria y de múltiples escalas de la entrada que captura, la información de sus colores y resolución, es decir, de su estilo, pero no la disposición global.

De nuevo, podemos visualizar la información capturada por capas de estilo construidas en diferentes capas de la red, dando lugar a una imagen que coincida con el estilo de representación de una determinada imagen de entrada. Las reconstrucciones a partir de las características del estilo producen versiones coloreadas de la imagen de entrada.

El hallazgo clave del algoritmo de transferencia de estilo es que las representaciones de contenido y estilo en la Red Neuronal Convolucional son separables. Esto permite manipular ambas representaciones independientemente para producir nuevas imágenes. Para demostrar este hecho, a continuación, se muestran algunas de las imágenes generadas mezclando el contenido y el estilo de representación de dos ortoimágenes aéreas de la misma área.

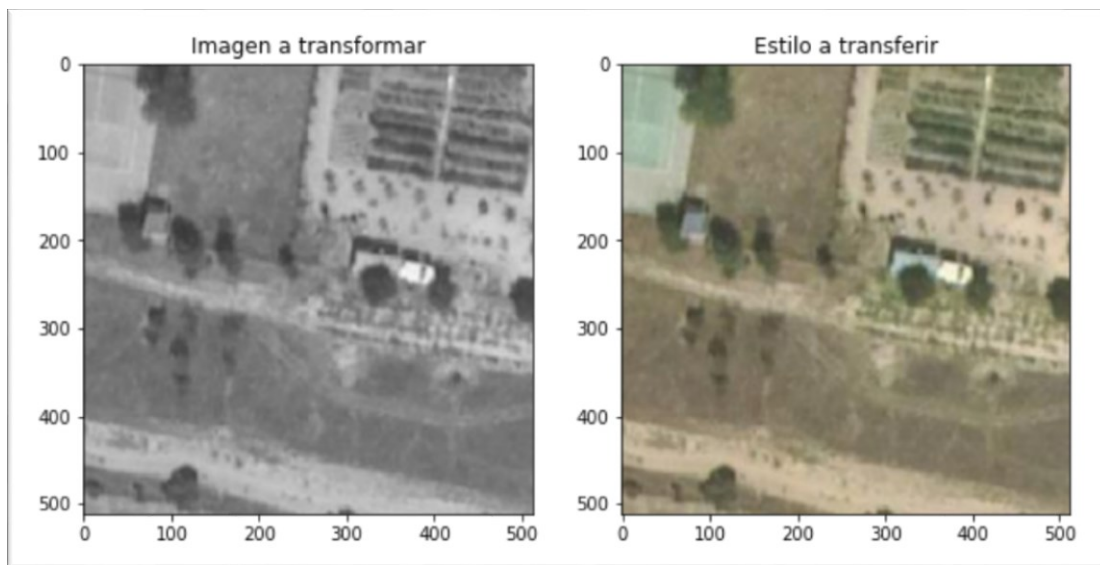


Figura 3.12. Imágenes de entrada utilizadas para la prueba de coloración mediante la transferencia de estilo con Deep Learning

Fuente: elaboración propia, a partir de datos del IGN

Las imágenes se sintetizan encontrando una imagen que coincida simultáneamente con el contenido de la ortoimagen a colorear (por ejemplo una tesela de 256x256 píxeles correspondiente a la hoja MTN50 0534 del PNOA de máxima actualidad, obtenida del centro de descargas del IGN y transformada en escala de grises mediante la extracción del canal L del espacio de color L^*a^b de la imagen original) y la representación de estilo de la ortoimagen correspondiente a ese área (la imagen original del PNOA de Máxima Actualidad). A continuación, puede verse el resultado de aplicar el estilo del PNOA a la misma imagen, pero en escala de grises.



Figura 3.13. Ampliación del mejor resultado de colorización obtenido mediante aplicación de la transferencia de estilo con Deep Learning

Fuente: elaboración propia, a partir de datos del IGN

Mientras que la disposición global de la ortoimagen original se conserva, los colores y los detalles que componen el escenario global son proporcionados por los datos de la imagen que contiene el estilo a transferir, en este caso procedente de la imagen original del PNOA.



Figura 3.14. Resultados de colorización obtenidos mediante la transferencia de estilo, extrayendo las capas de estilo y contenido a distintas profundidades de la red neuronal

Fuente: elaboración propia, a partir de datos del IGN

Las imágenes más atractivas visualmente se crean, normalmente, haciendo coincidir la representación de estilo hasta las capas más altas de la red.



Figura 3.15. Ampliación del mejor resultado de colorización obtenido mediante aplicación de la transferencia de estilo con Deep Learning a una imagen de El Retiro (Madrid), del Vuelo Americano Serie B

Fuente: elaboración propia, a partir de datos del IGN

Por supuesto, el contenido de la imagen y el estilo no se pueden aislar completamente. Al sintetizar una imagen que combina el contenido de una imagen con el estilo de otra, normalmente no existe una imagen que encaje perfectamente con ambas restricciones simultáneamente. Sin embargo, la función de pérdida que minimizamos durante la síntesis de la imagen contiene dos términos para el contenido y el estilo respectivamente, que están bien separados.

Por lo tanto, podemos regular sin problemas la generación de nuevas ortoimágenes haciendo hincapié en la reconstrucción del contenido o del estilo, como se aprecia en las imágenes incluidas a continuación.



Figura 3.16. Resultados de colorización obtenidos mediante la transferencia de estilo, extrayendo las capas de estilo y contenido a distintas profundidades de la red neuronal

Fuente: elaboración propia, a partir de datos del IGN

Si se enfatiza el estilo, resultará en imágenes que coincidan con la apariencia de la capa que lo aporta, dando una versión texturizada de la misma, pero sin mostrar prácticamente nada del contenido de la fotografía que se pretende colorear; esto supondría un problema en el caso de áreas en las que haya habido variaciones importantes entre la captura de una y otra ortoimagen. Cuando se enfatiza el contenido, se puede identificar claramente la fotografía, pero el estilo que se pretende transferir (el del PNOA) no está tan bien emparejado. Para un par específico de imágenes, se puede ajustar el equilibrio entre el contenido y el estilo para crear representaciones que sean lo más fieles posible al contenido de las ortoimágenes originales, con un estilo muy similar al que se pretende transferir.

Estos resultados se obtienen mediante la aplicación de un sistema neuronal artificial, que logra una separación del contenido de la imagen, permitiendo así fusionar el contenido de una capa en el estilo de cualquier otra capa, lo cual queda demostrado con los ejemplos expuestos en el presente epígrafe. Esto ha sido posible creando representaciones neuronales para el contenido y el estilo de una imagen a partir de las respuestas de las

características de las Redes Neuronales Profundas de alto rendimiento entrenadas en el reconocimiento de objetos, para lo cual se han utilizado modelos ya preentrenados como VGG16, VGG19, VGG22, ResNet, etc., siendo los mejores resultados los obtenidos mediante la aplicación del modelo VGG22. Los ejemplos mostrados en este epígrafe fueron todos generados sobre la base de la Red VGG22, una red neuronal convolucional que rivaliza con la actuación humana en un reconocimiento de objetos. Aun así, probablemente estos resultados pueden mejorarse notablemente con la aplicación de un modelo entrenado únicamente con datos procedentes de ortoimágenes aéreas, sin variar la metodología expuesta. Otro problema que presentan las redes que se apoyan en los modelos VGG es que no pueden adaptarse al entrenamiento con PatchGAN (ya que esta arquitectura no existía cuando fueron creadas las arquitecturas VGG), el cual reduce los tiempos de entrenamiento, y por lo tanto la adaptación de estos modelos a nuevos dominios resulta más costosa en términos computacionales.

La novedad presentada en este documento es que ésta es la primera demostración de la aplicación de este tipo de técnica en ortoimágenes aéreas, con todas las posibilidades que ello abre en el campo del trabajo cartográfico y la teledetección. Al ser novedosa esta aplicación, también lo son una serie de conclusiones que pueden extraerse de la misma, como el hecho de la mejora de la transferencia de estilo y los resultados obtenidos al trabajar con imágenes correspondientes a una misma localización o el hecho de que al partir de modelos preentrenados para la visión artificial, alimentados con imágenes de muchos tipos, pero no con ortoimágenes aéreas, los resultados serían mejorables si los modelos hubiesen sido expresamente entrenados con ortoimágenes aéreas, motivo por el cual se ha optado por el entrenamiento de modelos propios alimentados expresamente con los datos adecuados para optimizar los resultados al máximo posible y obtener representaciones de ortoimágenes aéreas en escala de grises con el estilo y características de la capa del PNOA Máxima Actualidad.

Anteriormente ya se habían hecho intentos de separación del contenido y el estilo, pero éstos fueron evaluados con imágenes de menor complejidad y nunca en ortoimágenes aéreas y, por lo tanto, sin aprovechar las ventajas que ofrece la geolocalización de estas y sin tener tampoco que hacer frente a los retos que suponen imágenes de esta complejidad. Pueden mencionarse como ejemplos previos las transferencias de estilos entre caracteres de diferente grafía o fuentes de texto distintas o imágenes de rostros, recogidos en los artículos de Tenenbaum et al. y de Elgammal et al.

Estos enfoques anteriores se basan, principalmente, en técnicas no paramétricas para manipular directamente la representación en píxeles de una imagen. Por el contrario, utilizando redes neuronales profundas entrenadas en el reconocimiento de objetos, se puede llevar a cabo la manipulación de características que representen explícitamente el contenido de alto nivel de una ortoimagen.

Las características de las redes neuronales profundas, entrenadas en el reconocimiento de objetos, han sido previamente utilizadas para el reconocimiento de estilos de una imagen; por ejemplo, en algunas de las aplicaciones de clasificación de obras de arte. El método propuesto de síntesis de ortoimágenes que mezclan contenido y estilo de diferentes capas proporciona una nueva herramienta, no sólo para la cartografía, sino para estudiar la percepción y la representación neuronal de la cartografía, el estilo y la apariencia de la ortoimagen aérea independiente del contenido en general.

Resulta fascinante que un sistema neuronal, que está entrenado para realizar una de las tareas computacionales básicas de la visión biológica, como es la detección de objetos, aprenda automáticamente a generar representaciones de imágenes que permiten la separación del contenido de las ortoimágenes de su estilo. La explicación podría ser que, al aprender reconocimiento de objetos, la red tiene que ser invariable a toda variación de imagen que preserve la identidad de objeto. Por lo tanto, la capacidad que

mostramos los humanos para procesos como la separación del contenido al analizar ortoimágenes aéreas del estilo y por ello, nuestra capacidad de crear mapas y admirar su belleza, podría ser principalmente fruto de las capacidades de inferencia de nuestro sistema visual.

3.6. Pruebas con el modelo DeOldify

Para esta prueba, se ha utilizado una arquitectura basada en las redes generativas antagónicas, conocida como NoGAN, y que está diseñada para hacer la conversión de imagen a imagen; es el modelo existente más parecido a los desarrollados durante esta investigación, pero ha sido diseñado para la restauración de fotografías y películas antiguas, no para su uso en cartografía. La mayor peculiaridad que presenta esta arquitectura es que se basa en la combinación del modelo convolucional conocido como ImageNet con el entrenamiento posterior de la red alimentada con imágenes en color y su equivalente en blanco y negro (el mismo diseño de entrenamiento utilizado para algunos de los modelos aquí desarrollados). El modelo DeOldify utiliza un generador U-Net preentrenado con las imágenes del modelo ImageNet. Cabe mencionar que en los modelos creados el generador utilizado también es un generador U-Net, pero entrenado directamente con datos procedentes de ortoimágenes aéreas en lugar de utilizar modelos preexistentes, para así obtener mejores resultados en el tipo de datos que se utilizan en el trabajo cartográfico. Esta metodología proporciona los beneficios del entrenamiento de las redes generativas antagónicas, mientras se invierte menos tiempo en el entrenamiento directo de ese tipo de red.

Al aplicar las redes generativas como complemento al modelo más tradicional VGG16, el entrenamiento de la Red Generativa Antagónica suple las carencias del modelo tradicional. Este tipo de arquitectura, a pesar de sus ventajas, no ha sido escogido para el desarrollo de los modelos propios para el trabajo con ortoimágenes, debido a que se basa en el modelo VGG16 preentrenado y entrenado posteriormente con los datos de ImageNet (los cuales no incluyen ortoimágenes), como se comentó anteriormente, no existen modelos preentrenados con datos de ortoimágenes, siendo la consecuencia de esta carencia que los inconvenientes superasen a las ventajas a la hora de utilizar la arquitectura NoGAN.

A continuación, se muestran algunos de los resultados obtenidos con este sistema al ser aplicado sobre teselas obtenidas mediante la conversión a escala de grises de las teselas procedentes de la hoja 0534 de la capa PNOA de máxima actualidad.

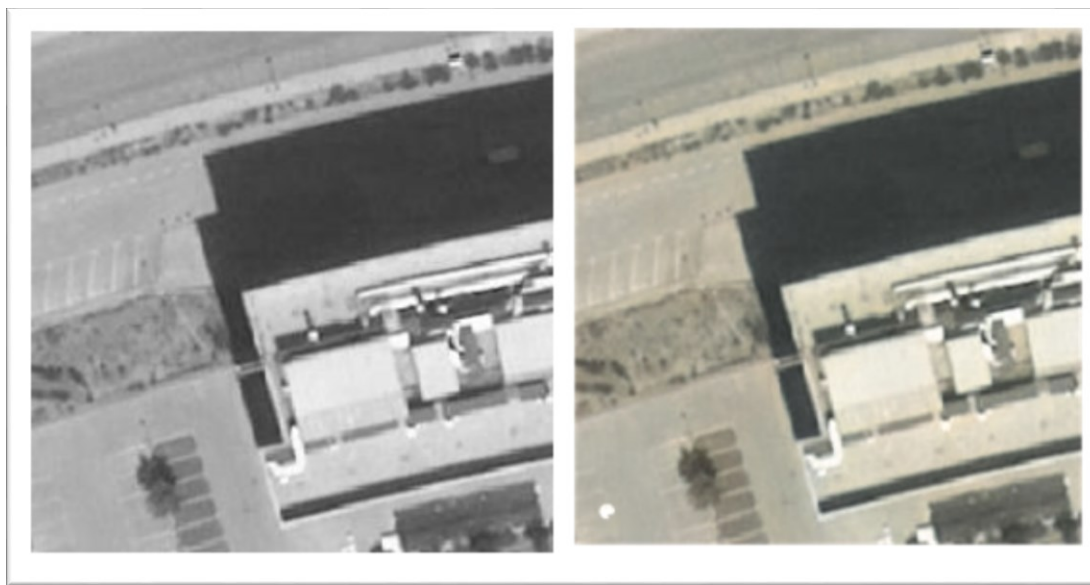


Figura 3.17. Imagen original e imagen coloreada mediante la red neuronal DeOldify
Fuente: elaboración propia, a partir de datos del IGN

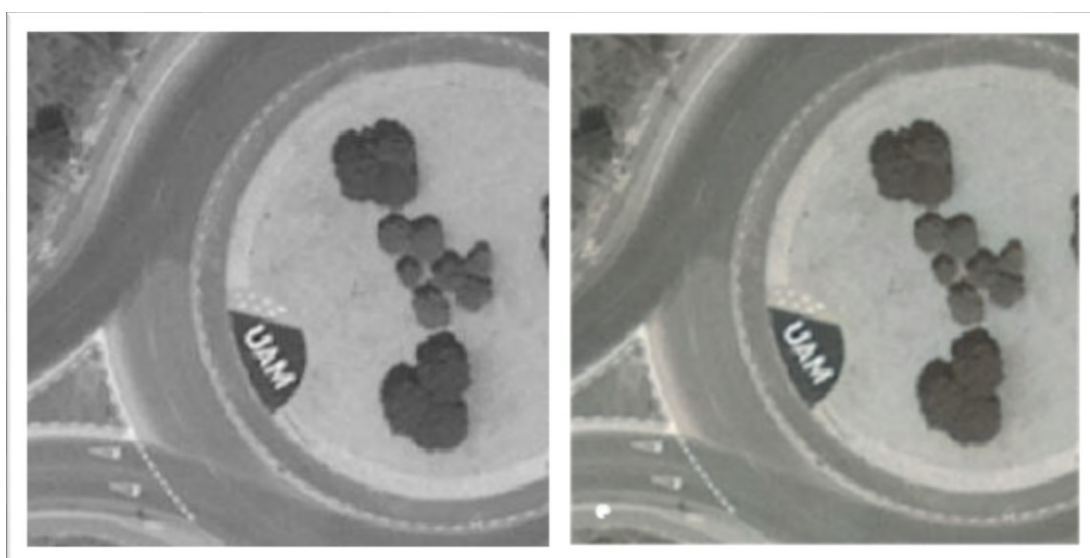


Figura 3.18. Imagen original e imagen coloreada mediante la red neuronal DeOldify
Fuente: elaboración propia, a partir de datos del IGN

A pesar de que los resultados, utilizando redes generativas antagónicas, son mucho mejores que con la simple transferencia de color, sigue existiendo el problema de colorear imágenes con las que no ha sido entrenada, como puede comprobarse principalmente en la tesela correspondiente al entorno urbano, ya que las imágenes habituales con las que la red ha sido entrenada, en ocasiones incluyen paisajes que podrían asemejarse a la ortoimagen aérea de un entorno rural, pero no se han incluido fotografías con vista cenital de una ciudad.

3.6.1. Conclusión de las pruebas con técnicas conocidas

Debido a los resultados poco satisfactorios obtenidos al aplicar las técnicas existentes para la coloración de imágenes en blanco y negro a la coloración de ortoimágenes aéreas para su uso en cartografía y teledetección, el problema de la coloración de imágenes para las aplicaciones anteriormente mencionadas, requiere nuevos planteamientos y técnicas, por lo que se ha optado por la creación de nuevas inteligencias artificiales, entrenadas expresamente para resolver los desafíos cartográficos que son objeto de esta tesis. Para ello, tras una extensa revisión bibliográfica de las distintas arquitecturas existentes en lo que a redes neuronales se refiere y tras la revisión de resultados de diferentes pruebas realizadas con las mismas, se ha llegado a la conclusión de que el mejor planteamiento es utilizar la arquitectura de las redes generativas, concretamente en un nuevo tipo de redes generativas propuesto por Goodfellow en 2014, conocido como redes generativas antagónicas y basar la coloración de las imágenes, no sólo en la identificación de los elementos de la imagen de origen, sino permitir que sea la propia inteligencia artificial la que recree una imagen plausible a partir de los datos de origen. Este nuevo planteamiento, permite afrontar todos los desafíos planteados en esta investigación con una única arquitectura, variando solamente los datos con los que son entrenados los

diferentes modelos, pudiendo, incluso, resolver varias cuestiones con un mismo modelo; esto es debido a que, al ser la propia inteligencia artificial la que crea una representación plausible en base a los datos suministrados, puede incluir color a la imagen, corregir defectos como el enfoque, los artefactos, deterioros y aumentar la resolución de la misma.



Figura 3.19. Ejemplo del resultado de la aplicación del modelo propio DeepMapsOptimiza, el cual genera una versión de la imagen de entrada en la que se corrige el desenfoque y se elimina el ruido

Fuente: elaboración propia, a partir de datos del IGN

Básicamente, la inteligencia artificial imagina una imagen con las características que se han utilizado para su entrenamiento (Datos de ortoimágenes del PNOA de Máxima Actualidad), en base a la imagen suministrada para tratar; por ejemplo, si se le suministra una imagen del Vuelo Americano Serie B, en la cual aparece un bosque, generará una imagen que podría ser confundida con las imágenes del PNOA de Máxima Actualidad, pero en la que aparece el bosque de la imagen original, con la distribución de elementos presente cuando se tomó la imagen del vuelo americano en los años 50.

3.7. Modelos propios



Figura 3.20. Ejemplo de ortofotografía aérea coloreada con la inteligencia artificial DeepMapsColor19

Fuente: elaboración propia, a partir de datos del IGN

Como ya se ha mencionado anteriormente, se ha podido comprobar cómo, a pesar de que algunos de los algoritmos y redes neuronales existentes, desarrolladas para el tratamiento de imágenes, pueden adaptarse para el desempeño de labores cartográficas. Los resultados obtenidos podrían mejorarse si se entrenan las redes específicamente con datos de ortoimágenes aéreas y satelitales. Debido a esto, se han entrenado durante varios meses algunos modelos de redes neuronales para su aplicación específica en labores cartográficas; para obtener los resultados que se muestran a continuación, ha sido necesaria una profunda revisión bibliográfica y la realización de numerosas pruebas hasta llegar a establecer determinados principios que permiten sentar las bases de la aplicación de la inteligencia artificial en el procesamiento de imágenes en cartografía.

La creación de estos modelos tiene como meta el demostrar la validez de las hipótesis planteadas y servir como demostración de las posibilidades que

ofrece esta tecnología aplicada al mundo de la cartografía, así como de la validez de las bases planteadas para su utilización en este campo.

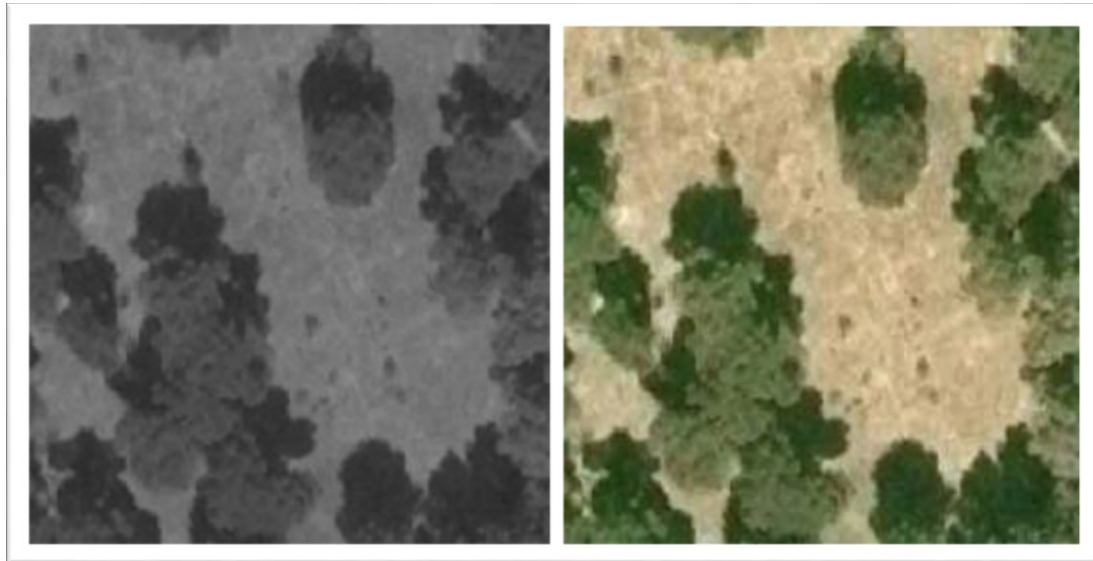


Figura 3.21. Ejemplo de ortofotografía aérea restaurada y coloreada con una de las inteligencias artificiales creadas (DeepMapsColor19)
Fuente: elaboración propia, a partir de datos del IGN

En los siguientes subepígrafes se muestran algunos de los modelos entrenados, que han proporcionado mejores resultados en el campo de la coloración automatizada de imágenes, así mismo se comentan también algunos detalles del entrenamiento, como los datos con los que han sido entrenados y el número de iteraciones realizadas durante el entrenamiento. También debe aclararse que todos los modelos entrenados tienen una misma arquitectura, la cual ya ha sido expuesta en el capítulo 2, junto con los detalles del hardware donde este entrenamiento ha sido ejecutado.

Los datos originales con los que se han alimentado los modelos han sido extraídos de capas suministradas por el IGN, mediante el software de tratamiento de datos geográficos QGIS, generando así teselas de 256x256 píxeles con las que se ha trabajado posteriormente.

Para la generación de los dataset de entrenamiento de las redes de colorización elaboradas, se ha recurrido al tratamiento de las imágenes originales de varias capas, aunque la mayoría proceden del PNOA y del Sentinel 2, mediante un script de Python, expresamente creado para el procesamiento masivo de estas imágenes. Este script genera imágenes a las que se les ha quitado la información correspondiente al color mediante la conversión de estas al espacio de color $L^*a^*b^*$, extrayendo el canal L de forma independiente y guardándolo como una nueva imagen. Mediante otro script se ha creado una composición de la imagen original y la imagen tratada para el entrenamiento, dando como lugar imágenes de 512x256 píxeles, con las que se ha realizado la ingesta de datos para el entrenamiento de los modelos. Se ha optado por unir las dos imágenes en un solo archivo con extensión jpg para reducir los tiempos de envío y de procesamiento, debido a que es más ligera que otros formatos de imagen.



Figura 3.22. Ejemplo de imagen de 512x256 píxeles generada para entrenar los modelos en la colorización de ortoimágenes aéreas

Fuente: elaboración propia, a partir de datos del IGN

3.7.1. DeepMapsColor19

Esta red ha sido entrenada con 10.000 imágenes seleccionadas de forma aleatoria de la totalidad del territorio español, mediante un script de Python desarrollado para tal fin, a partir de ortoimágenes aéreas del PNOA de Máxima Actualidad. Para crear los datos de entrenamiento se han utilizado teselas de 256x256 píxeles, obtenidas a partir de ortoimágenes del PNOA, con un nivel de zoom de 19 (escala aproximada de 1:1.000). El número de iteraciones durante las que ha sido entrenado este modelo asciende a 30.000, lo cual ha supuesto meses de procesamiento.

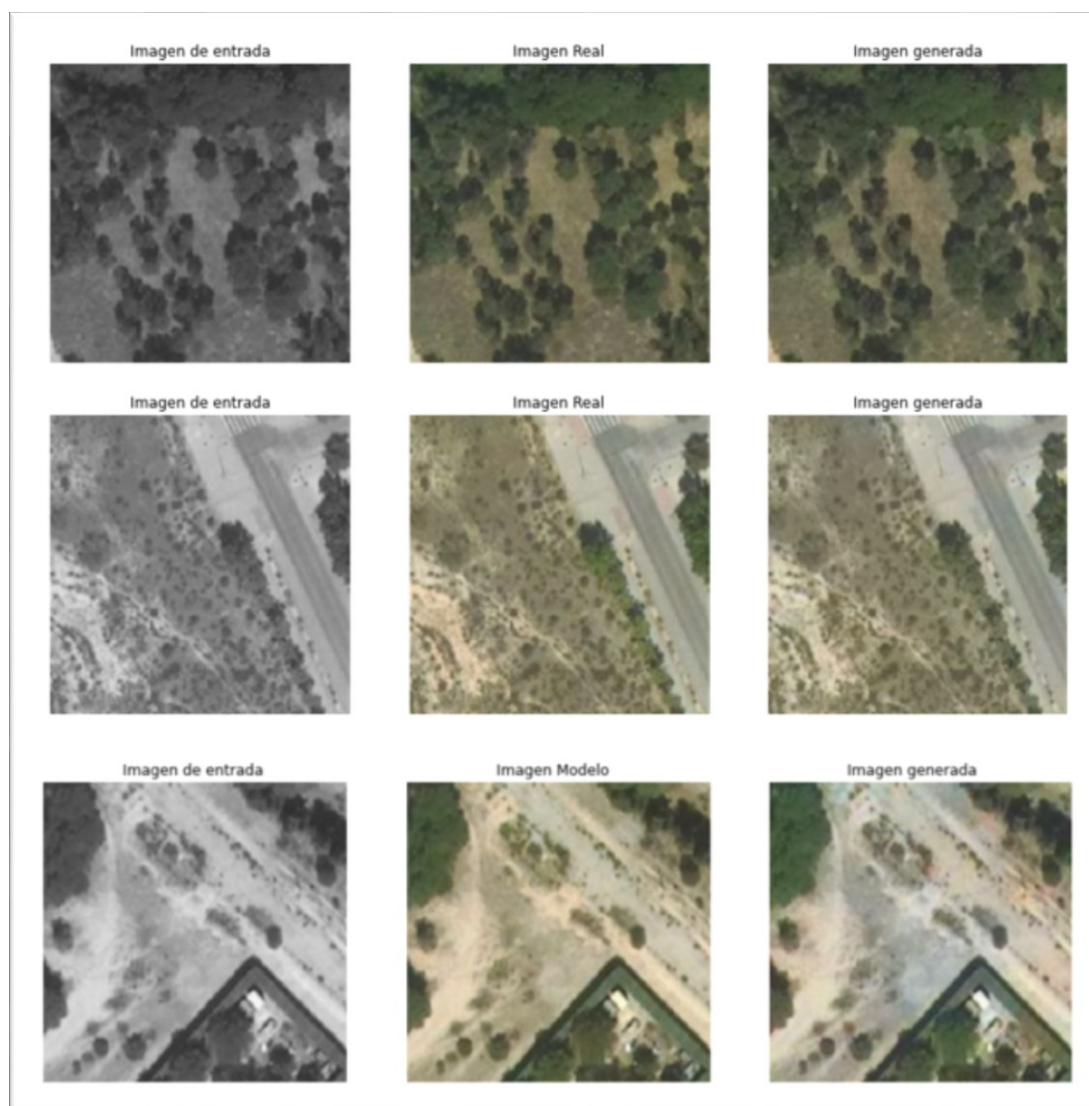


Figura 3.23. Resultados de la aplicación del modelo DeepMapsColor19
Fuente: elaboración propia, a partir de datos del IGN

En la siguiente muestra puede verse como el modelo desarrollado, permite no sólo colorear las imágenes de forma realista, sino que, al generar una nueva representación, cuando durante el entrenamiento dispone de información suficiente de los elementos a representar, es capaz de generar una imagen que posee, incluso, mayor nitidez y colores más vivos que la original, corrigiendo también defectos en la coloración original, como se muestra en el siguiente ejemplo.

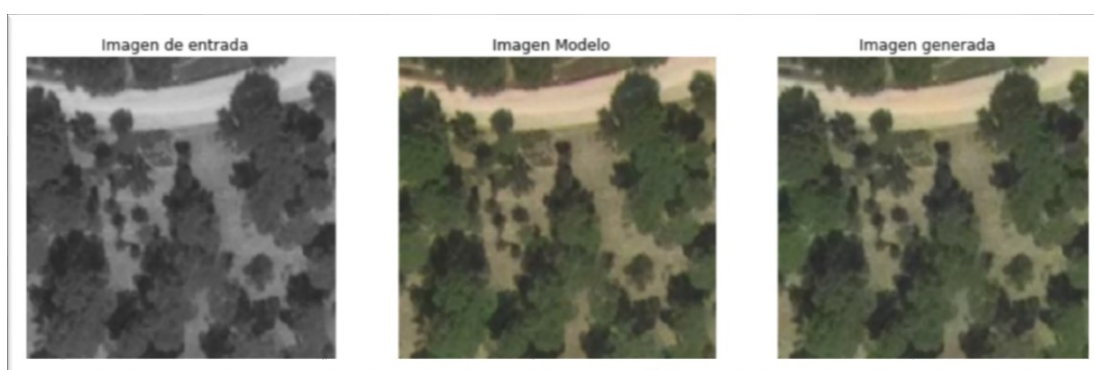


Figura 3.24. Ejemplo de resultado de la aplicación del modelo DeepMapsColor19.
Fuente: elaboración propia, a partir de datos del IGN

Este modelo ha sido entrenado para que funcione preferiblemente con imágenes que tengan un nivel de zoom de 19, es decir una escala aproximada de 1:1.000; sin embargo, devuelve también muy buenos resultados con niveles de zoom no demasiado alejados de éste.



Figura 3.25. Ejemplo de resultados obtenidos con el modelo DeepMapsColor19
Fuente: elaboración propia, a partir de datos del IGN

El entrenamiento necesario para la creación de esta inteligencia ha sido el más costoso de cuantos se han realizado, pero demuestra la suposición de que cuanto mayor es el dataset de entrenamiento y mayor número de iteraciones se realizan, mejores son los resultados, siempre y cuando no se sobreentrene el modelo. El haber creado el dataset de entrenamiento con las ortoimágenes originales del PNOA y esas mismas ortoimágenes en escala de grises, responde a la necesidad de aportar a la inteligencia artificial un ejemplo de cómo deben añadirse colores a los distintos elementos presentes en las ortoimágenes, es decir, para enseñarle que partiendo de una imagen en escala de grises debe generar una imagen a color.

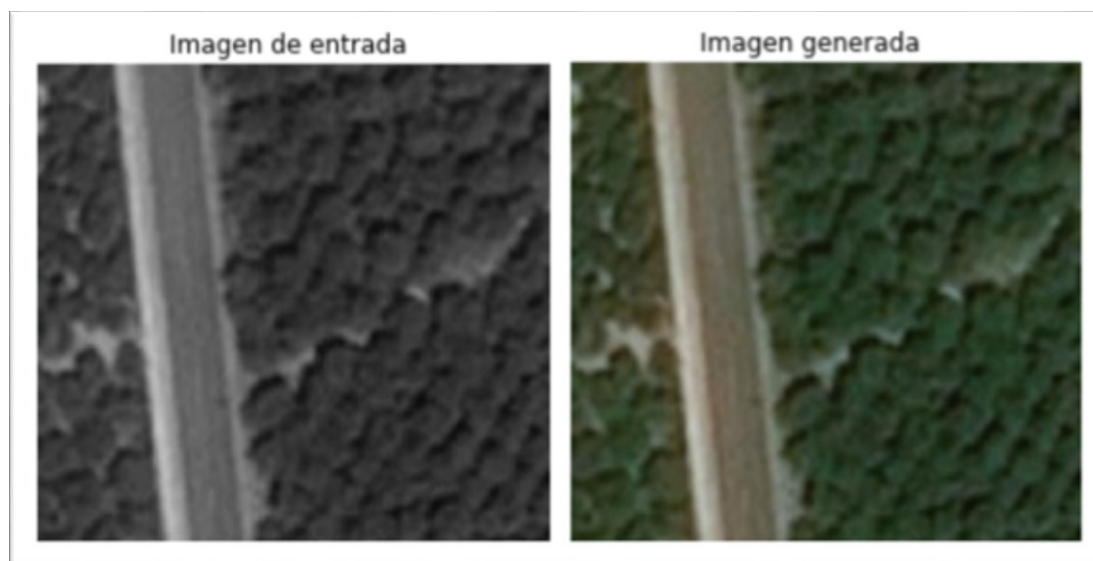


Figura 3.26. Ejemplo de resultados obtenidos con el modelo DeepMapsColor19
Fuente: elaboración propia, a partir de datos del IGN

3.7.2. DeepMapsColor15

Este modelo se ha entrenado con 1.000 imágenes seleccionadas de forma aleatoria, partiendo de los datos de las hojas correspondientes a la Comunidad de Madrid, durante 6.000 iteraciones. Para crear los datos de entrenamiento se han utilizado teselas de 256x256 píxeles, obtenidas a partir de ortoimágenes del PNOA, con un nivel de zoom de 15 (escala aproximada de 1:15.000) y versiones en escala de grises de estas obtenidas mediante un script de Python para componer los pares de imágenes necesarios para el entrenamiento de la red.

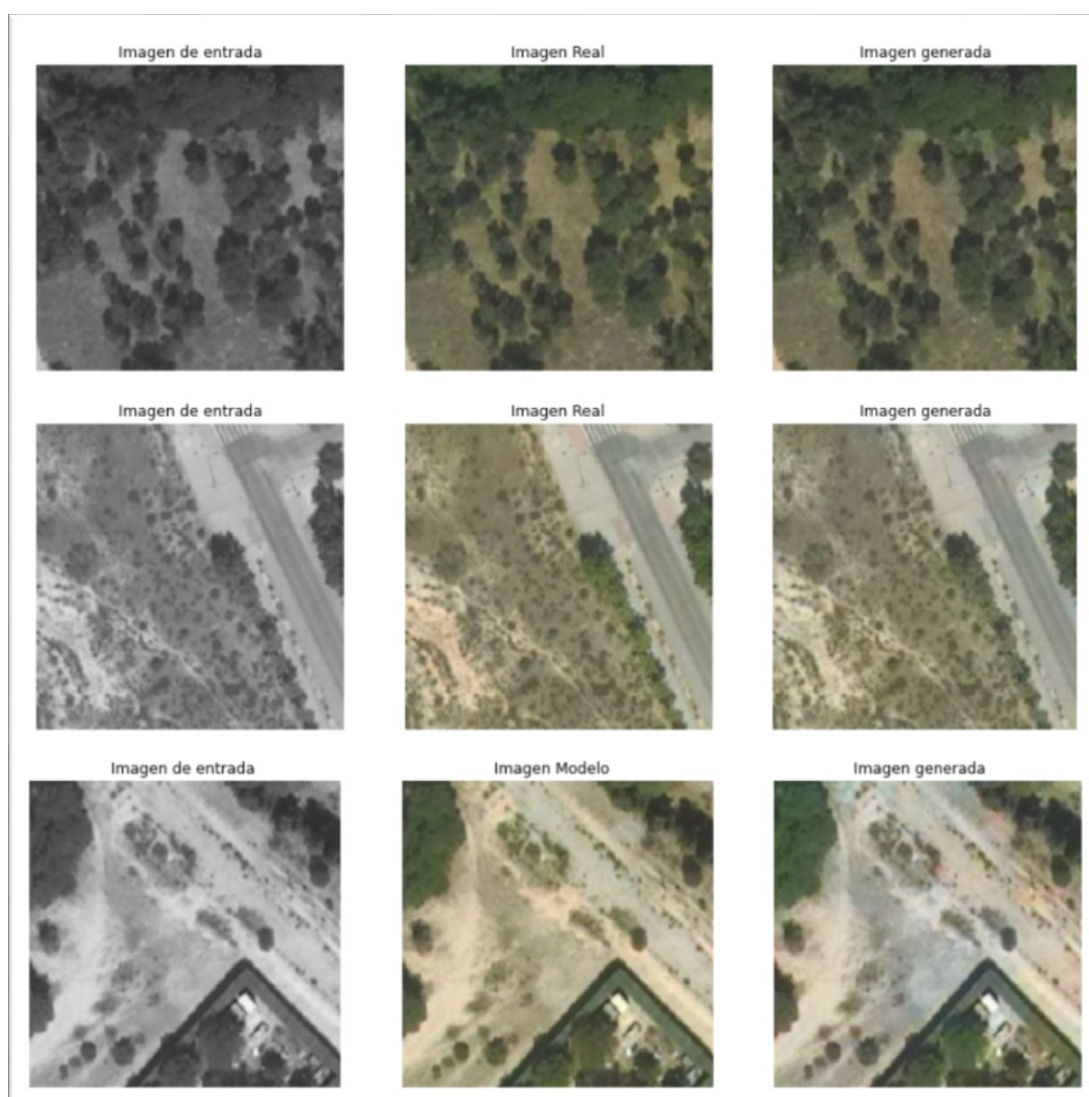


Figura 3.27. Resultados de la aplicación del modelo DeepMapsColor15
Fuente: elaboración propia, a partir de datos del IGN

Una de las ventajas de la utilización de Redes Neuronales Generativas es que, cuanto más aprenden, mejores son los resultados, siempre y cuando no sean sobreentrenadas y los datos con los que se ha realizado el entrenamiento sean de calidad. Durante el entrenamiento de las inteligencias artificiales han podido observarse comportamientos emergentes para los que las redes no habían sido entrenadas, como la creación de nuevas categorías y gradientes en los mapas temáticos o la mejora de la resolución y el enfoque, además de la eliminación de defectos, tal y como puede comprobarse en los distintos ejemplos expuestos en este documento, especialmente en aquellos en los que se han aplicado las inteligencias artificiales a datos de ortofotografías antiguas, como la mostrada a continuación.



Figura 3.28. Ejemplo de coloración del Vuelo Americano Serie B (1956-1957) mediante el modelo DeepMapsColor15 (Parque de la Bombilla y zona deportiva Campus Sur de la Universidad Complutense de Madrid)

Fuente: elaboración propia, a partir de datos del IGN

A pesar de haber sido entrenadas con datos del PNOA editados, las redes neuronales creadas para asignar color a las imágenes en blanco y negro funcionan perfectamente en datos de otras fuentes, tal y como se muestra en la ortomagen anterior, en la que se ha aplicado la inteligencia artificial DeepMapsColor15 a una ortomagen procedente del Vuelo Americano Serie B.

3.7.3. DeepMapsColor17

Esta red ha sido alimentada con pares de imágenes generadas a partir de teselas de 256x256, generadas a partir de las hojas del PNOA de Máxima Actualidad correspondientes a la Comunidad de Madrid, con una escala aproximada de 1:4.000. El entrenamiento se realizó con 500 imágenes sobre la red ya preentrenada DeepMapsColor15, debido a lo cual sólo fueron necesarias 2.000 iteraciones para obtener estos resultados.

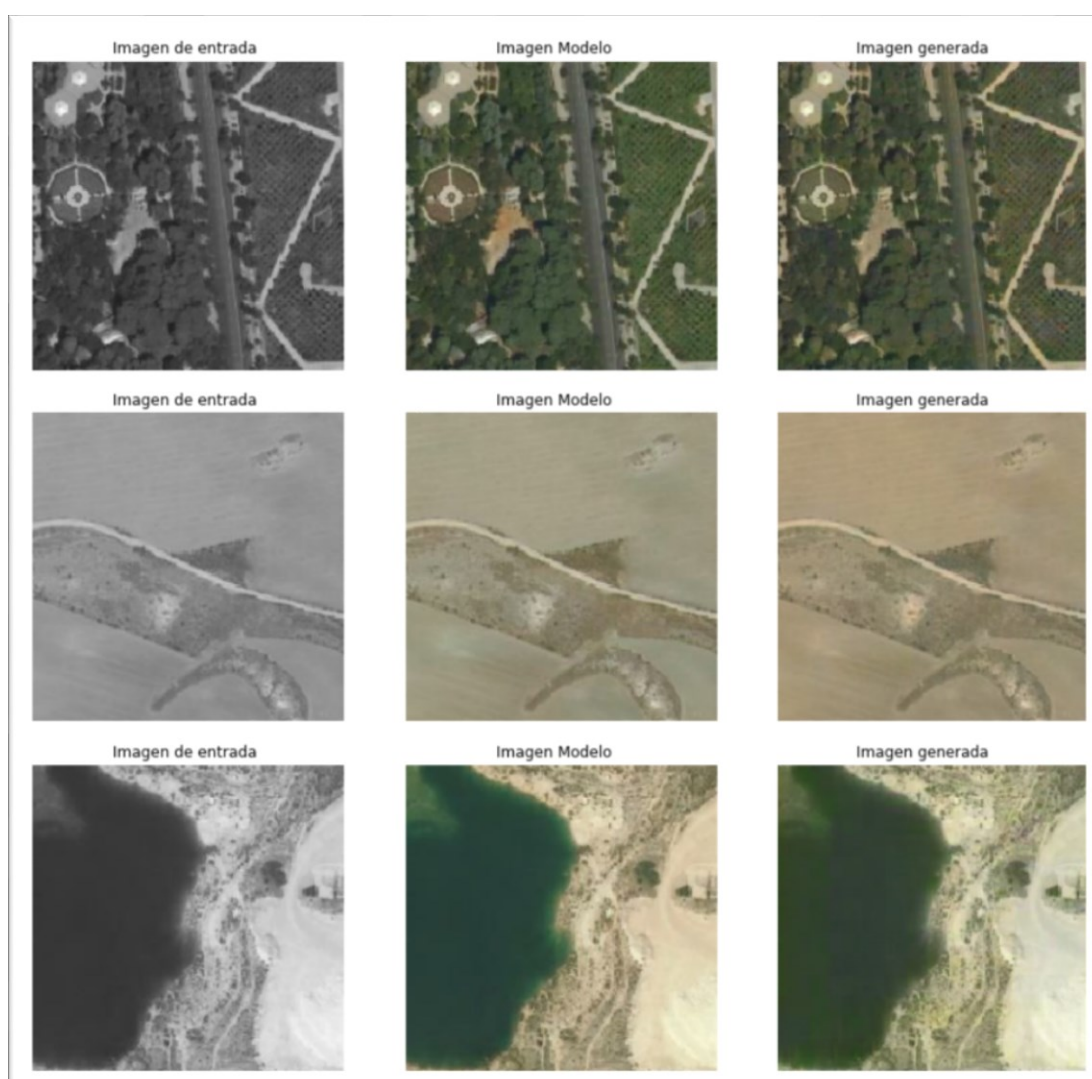


Figura 3.29. Resultados de la aplicación del modelo DeepMapsColor17
Fuente: elaboración propia, a partir de datos del IGN



Figura 3.30. Ejemplo de coloración del Vuelo Americano Serie B (1956-1957) restaurada con el modelo DeepMapsColor17 (Lago de la Casa de Campo)
Fuente: elaboración propia, a partir de datos del IGN

3.7.4. DeepMapsColorMultizoom

Esta red ha sido entrenada con 500 imágenes durante 2.800 iteraciones, obtenidas a partir de las teselas de 256x256 píxeles, procedentes de ortoimágenes satelitales del Sentinel 2 a distintos niveles de zoom, que van del 1 al 10 (escalas aproximadas hasta 1:500.000) seleccionadas de forma aleatoria de todo el territorio nacional.

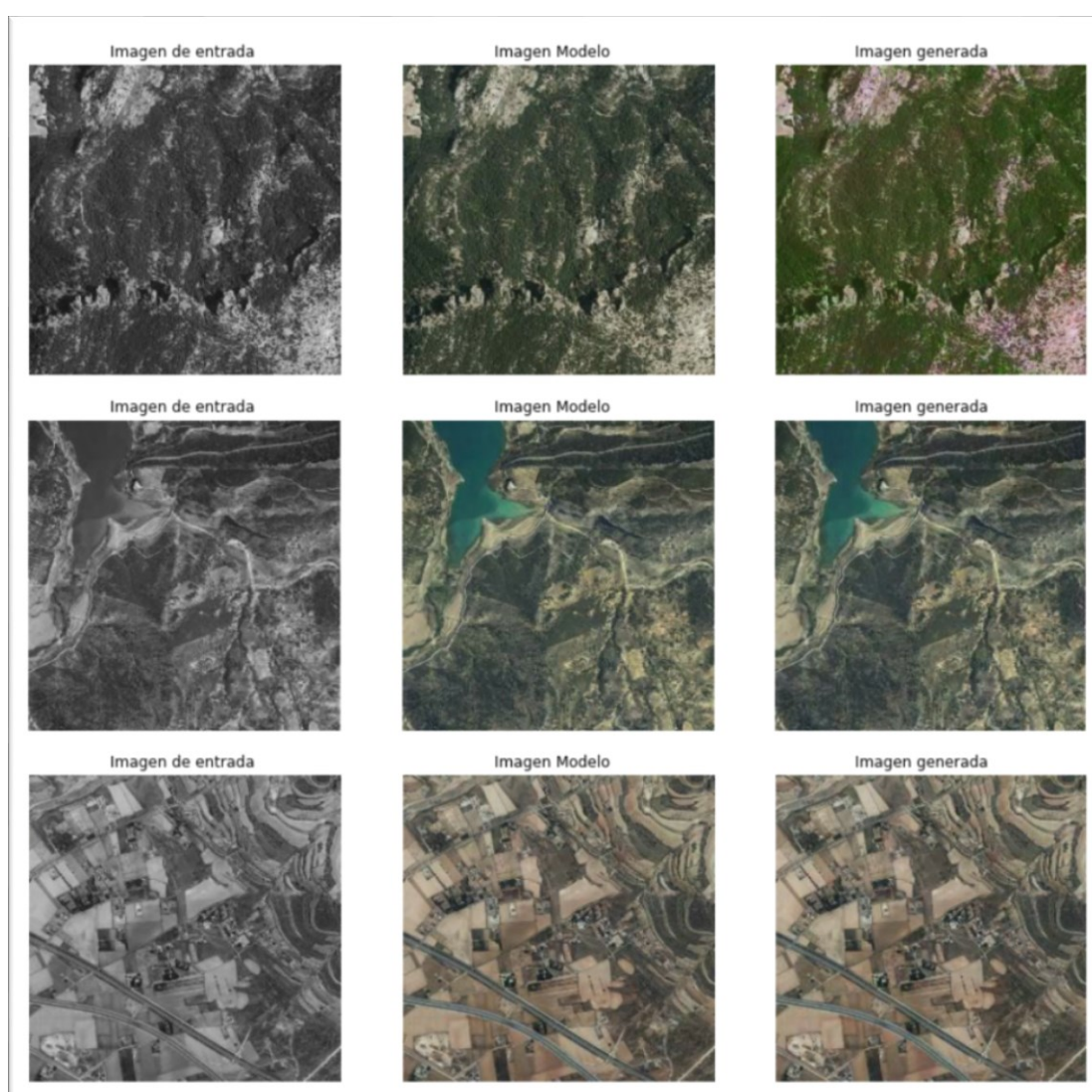


Figura 3.31. Resultados producidos por el modelo DeepMapsColorMultizoom en relación con las imágenes originales (Imagen Modelo)

Fuente: elaboración propia, a partir de datos del IGN

El modelo DeepMapsColorMultizoom ha sido entrenado para que realice la coloración automática de imágenes en varias escalas diferentes; resulta más versátil que las diseñadas específicamente para distintos niveles de zoom, pero los resultados obtenidos, a pesar de ser verosímiles y mucho mejores que los obtenidos con otros modelos o técnicas (ya que las inteligencias artificiales aquí presentadas son las primeras creadas específicamente para realizar esta labor en ortoimágenes aéreas), no llegan a ser tan fieles a la realidad como los que se han obtenido con las otras redes entrenadas para niveles de zoom más específicos.



Figura 3.32. Ejemplo de resultado obtenido con el modelo DeepMapsColorMultizoom con respecto a la imagen original (Imagen Modelo) procedente de la capa del PNOA de máxima actualidad

Fuente: elaboración propia, a partir de datos del IGN

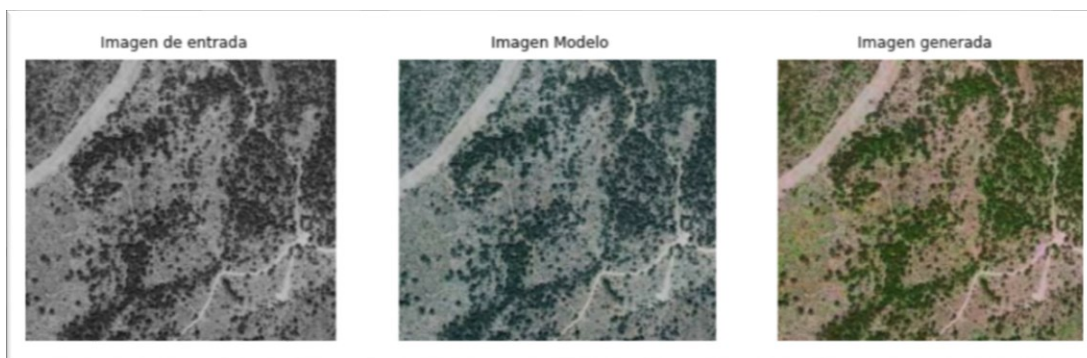


Figura 3.33. Ejemplo de resultado obtenidos con el modelo DeepMapsColorMultizoom con respecto a la imagen original (Imagen Modelo) procedente de la capa del PNOA de Máxima Actualidad

Fuente: elaboración propia, a partir de datos del IGN

Las imágenes superiores muestran los resultados obtenidos con este modelo en distintas resoluciones. Como puede comprobarse en los anteriores ejemplos, el modelo, una vez entrenado, es capaz incluso de mejorar la coloración de las imágenes originales con las que fue entrenado.

3.8. Conclusiones

Como se ha podido comprobar en las pruebas realizadas, los modelos entrenados específicamente con ortoimágenes aéreas son más eficaces al realizar la tarea de coloración de ortoimágenes aéreas, incluso aquellos que han sido entrenados para poder trabajar a varios niveles de zoom, que los métodos ya existentes para la coloración de imágenes de aplicación en el ámbito fotográfico. Esto se debe, sin duda, a que los algoritmos y modelos existentes para la colorización de imágenes tenían un propósito más general y no habían sido creados para el su utilización en cartografía. Esta cuestión se repite con el resto de los métodos utilizados actualmente en labores cartográficas, ya que hasta ahora la cartografía se ha limitado a utilizar métodos desarrollados para el tratamiento de imágenes, pero no creados específicamente para el trabajo cartográfico. Es por ello por lo que los modelos y conclusiones obtenidas con la presente investigación suponen un aporte muy valioso al mundo de la cartografía.

4. Resolución y enfoque

Para recuperar o restaurar imágenes de alta resolución, a partir de imágenes de baja resolución, existen numerosos métodos de mejora de imagen, los cuales incluyen reducción de ruido, aumento de resolución y ajustes de color, cuestiones que son analizadas en este documento mediante la aplicación de Deep Learning con redes generativas antagónicas condicionales.

En este capítulo, el objetivo principal es reconstruir una imagen de alta resolución al escalar la imagen de baja resolución, de modo que no se pierda el detalle de la textura en las imágenes reconstruidas, teniendo también en cuenta el coste de computación necesario y la viabilidad de cada uno de los métodos para el tratamiento masivo de datos procedentes de ortoimágenes aéreas, necesario para realizar las labores cartográficas que son el foco de este estudio.

Hace ya bastante tiempo que se vienen utilizando distintas técnicas de interpolación para aumentar la resolución de las imágenes, sin embargo, los resultados obtenidos por estas técnicas difieren entre sí y no siempre son adecuados. Los métodos más sofisticados explotan las similitudes internas de una imagen determinada o utilizan conjuntos de datos de imágenes de baja resolución y sus contrapartes de alta resolución para aprender, de manera efectiva, un mapeo entre ellas. Entre las distintas técnicas existentes, los métodos de aumento de resolución basados en el aprendizaje profundo han empezado a ofrecer en los últimos años mejores resultados a la hora de obtener imágenes optimizadas, tal y como se muestra en los resultados de las pruebas realizadas en este capítulo.

Por otra parte, se han creado nuevos modelos, tanto para corregir el desenfoque como para mejorar la resolución (entre otros), basados en redes generativas antagónicas condicionales con la arquitectura image-to-image, presentada en 2016.



Figura 4.1. Ejemplo de resultados obtenidos con el modelo DeepMapsDespixela
Fuente: elaboración propia, a partir de datos Wingtra

Como sucedía en el caso de la coloración de las imágenes en blanco y negro, las nuevas tecnologías pueden permitirnos la aplicación de las redes neuronales para mejorar el enfoque y la resolución de las imágenes, completando información ausente de forma mucho más inteligente que los algoritmos anteriormente utilizados. En el siguiente epígrafe se muestra un análisis de distintos algoritmos de interpolación y redes neuronales aplicados a la mejora de la resolución de las ortoimágenes aéreas.

4.1. Técnicas clásicas de interpolación para mejorar la resolución

En cartografía digital, la interpolación bidimensional ya ha sido ampliamente utilizada en el pasado. En este tipo de interpolación la intensidad de un píxel de la matriz de muestreo más densa, que no está disponible en la matriz de muestreo original, se estimará a partir de los píxeles de la matriz de muestreo original, de manera que todos los píxeles (tanto los nuevos como los antiguos) se colocarán en la misma superficie construida por la función de ajuste bidimensional.

En este epígrafe se realiza una breve descripción y la comparativa de algunos métodos de interpolación habituales para la mejora de la resolución, entre ellos los más habituales son nearest neighbor, bilineal y bicúbica, todos ellos ampliamente utilizados en el procesamiento de imágenes desde hace años.

Debido a su simplicidad, en comparación con sistemas de Deep Learning, el procesado de las imágenes resulta mucho más rápido con estos sistemas, pero por supuesto no llegan a alcanzarse resultados como los obtenidos con los últimos métodos que utilizan redes neuronales y, aún menos, los obtenidos con las redes entrenadas durante esta investigación. Esta comparación de resultados entre métodos clásicos y los aquí propuestos puede realizarse revisando los algoritmos incluidos en los primeros epígrafes, correspondientes a las técnicas existentes, y los resultados de los últimos epígrafes de este capítulo, correspondientes a los modelos entrenados durante las investigaciones llevadas a cabo para la elaboración de este documento.

A continuación, se analizan los métodos de interpolación existentes para aumentar el tamaño de la imagen, reduciendo la pixelación de la misma en la medida de lo posible.

4.1.1. Nearest Neighbor o vecino más próximo

Esta interpolación es la forma más simple y rápida de reducir o ampliar una imagen. Sin embargo, los resultados de este método son bastante inexactos. El método Nearest Neighbor puede ser de alguna utilidad sólo si se aumenta el tamaño de la imagen original en una cantidad exacta y concreta como 4 o 16 veces el tamaño de un original; esto es así debido a que los píxeles son cuadrados, y cuando se aumenta la altura de una imagen en un factor de dos 2, también se aumenta el ancho en ese mismo factor de 2. Por lo tanto, el doble de la resolución de un archivo de imagen significa 4 veces su tamaño original.



Figura 4.2. Resultado de la interpolación Nearest Neighbor aplicada a una ortoimagen en color
Fuente: elaboración propia, a partir de datos de Wingtra

Este método no realiza cálculos o promedios, sino que simplemente repite el color del píxel en el borde del original; esto provoca resultados poco atractivos e inexactos, especialmente si se aumenta el tamaño de una imagen por una cantidad fraccional, o una cantidad que se encuentra entre dos enteros, donde incluso empeora el aspecto con respecto a la imagen original. Este método no incluye información nueva, simplemente hace la imagen más grande; por lo tanto, no sirve a nuestro objetivo que es el de eliminar el pixelado de las teselas manteniendo su tamaño. A continuación, se muestra el resultado de este algoritmo sobre una de las teselas en escala de grises pixelada.

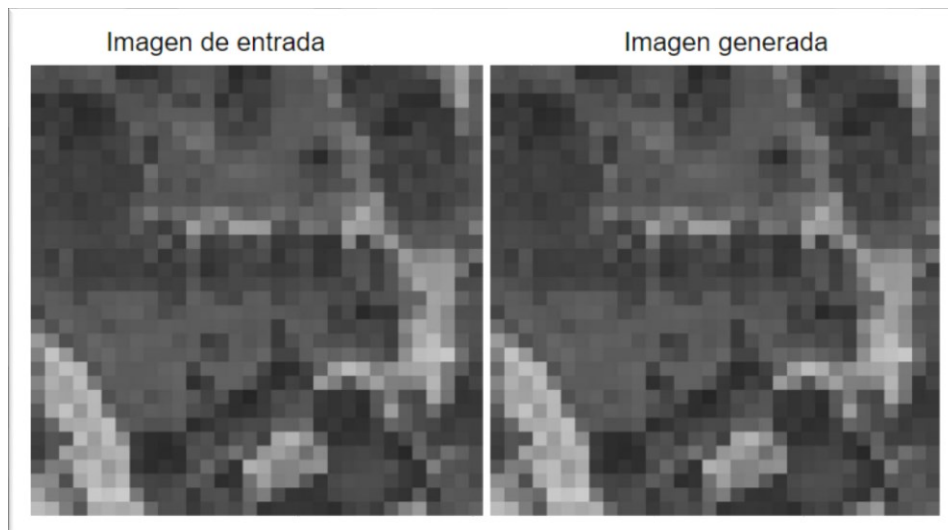


Figura 4.3. Resultado de la interpolación Nearest Neighbor aplicada a una ortoimagen en escala de grises

Fuente: elaboración propia, a partir de datos del IGN

El resultado de esta interpolación claramente produce una imagen pixelada. Además, no introduce ningún dato nuevo.

4.1.2. Interpolación bilineal

Cuando se aumenta el tamaño de una imagen también se puede usar la interpolación bilineal. La interpolación bilineal es menos intensiva en cuanto al procesamiento que la bicúbica porque hace un análisis menor de los píxeles antes de crearlos. Este método produce una imagen más suave que el Nearest Neighbor, pero los resultados de las transiciones nítidas como los bordes no son ideales porque los resultados son una media ponderada de los 2 píxeles circundantes.



Figura 4.4. Ejemplo de aplicación de la interpolación bilineal aplicada a una ortoimagen en color

Fuente: elaboración propia, a partir de datos de Wingtra

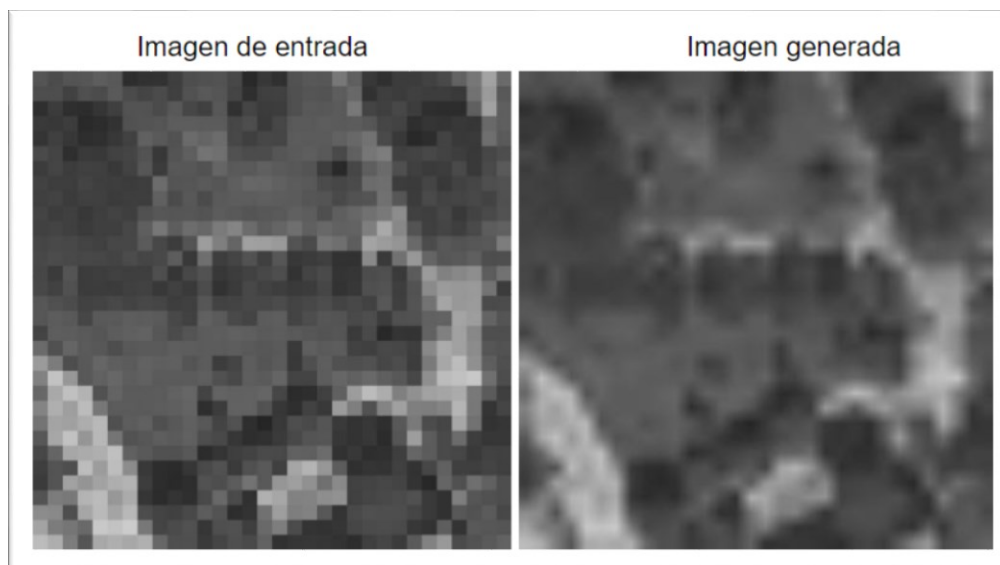


Figura 4.5. Ejemplo de aplicación de la interpolación bilineal aplicada a una ortoimagen en escala de grises

Fuente: elaboración propia, a partir de datos del IGN

El resultado de esta interpolación podría no ser obvio con imágenes muy pequeñas. Sin embargo, los nuevos píxeles y colores de los píxeles en una imagen en la que se ha utilizado la interpolación bilineal no serán tan fieles a la imagen original, ya que se realiza un buen análisis, pero no exhaustivo, para obtener los datos a promediar.

4.1.3. Interpolación Bicúbica

La interpolación bicúbica es el más sofisticado y elegante de los métodos de redimensionamiento incluidos en los programas habituales de tratamiento de imágenes. Este método también requiere mayor potencia de procesamiento que los dos anteriores, pero, a pesar de ser la interpolación más costosa en términos de computación, sigue siendo más ligera y eficiente que los métodos basados en redes neuronales.

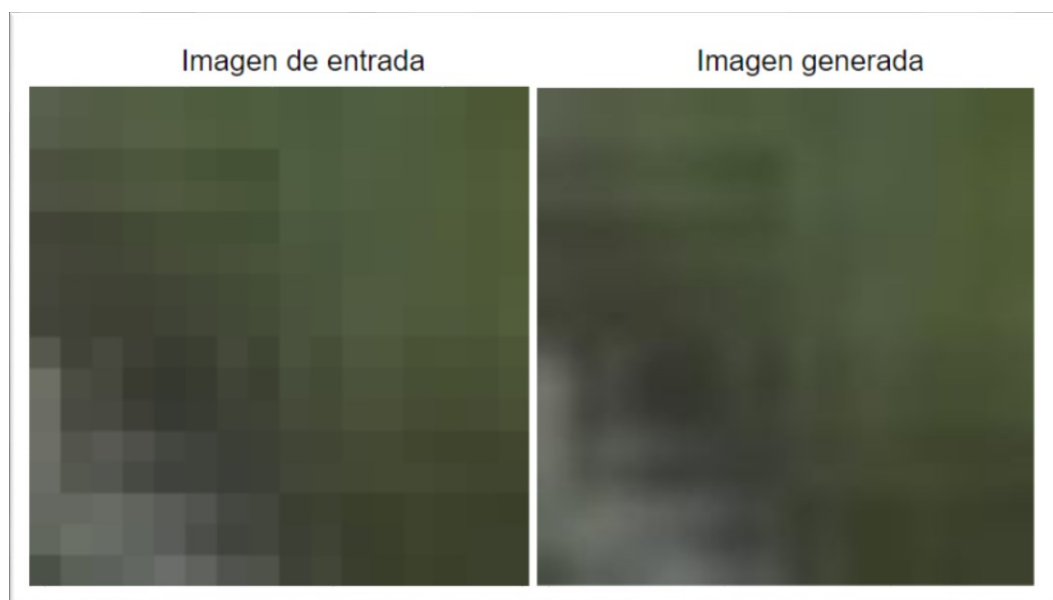


Figura 4.6. Ejemplo de aplicación de la interpolación bicúbica aplicada a una ortoimagen en color

Fuente: elaboración propia, a partir de datos de Wingtra

Lo que se puede concluir de la interpolación bicúbica es que el redimensionamiento de la ortoimagen hace que los píxeles incluidos en la imagen procesada aparezcan un poco borrosos.

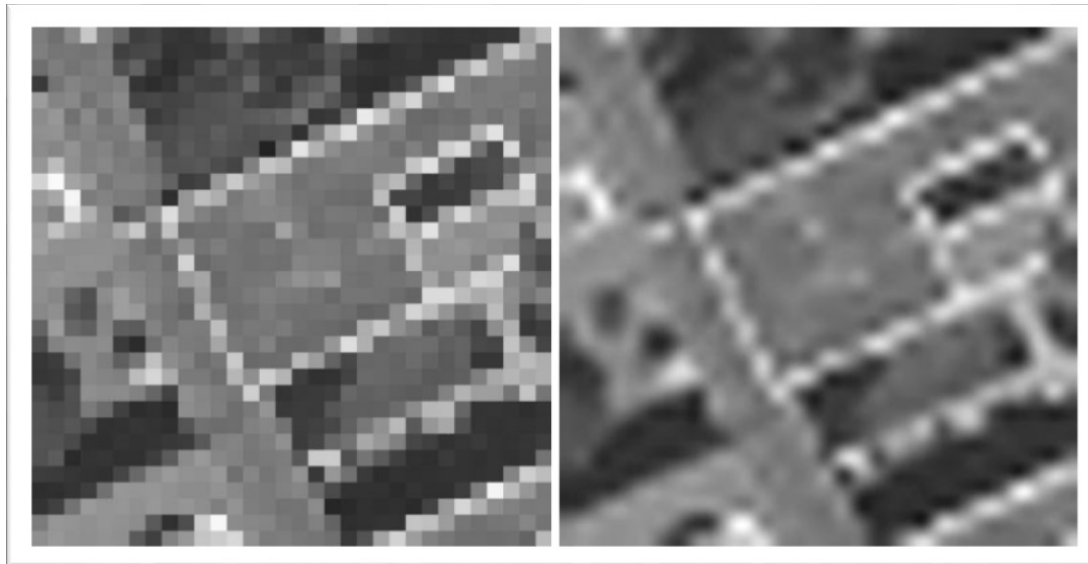


Figura 4.7. Ejemplo de aplicación de la interpolación bicúbica
Fuente: elaboración propia, a partir de datos del IGN

Como puede apreciarse en los resultados, la interpolación bicúbica produce una imagen más nítida que los dos métodos anteriores. Este método equilibra bastante bien el tiempo de procesamiento y la calidad de la producción, es por ello por lo que se ha incluido en el desarrollo del software para tratamiento y visualización de datos LiDAR (MEL), ya que mejora en gran medida los resultados, permitiendo detectar estructuras con mayor facilidad y sin consumir un tiempo excesivo en el procesamiento de las nubes de puntos.

Desde un punto de vista matemático, la diferencia entre la interpolación bilineal y la bicúbica es que la interpolación bilineal no pondera el promedio de las muestras que inventa a partir de los datos de imagen existentes, y busca en sólo dos direcciones, vertical y horizontal, para obtener nuevos colores y/o nuevos píxeles.

Al aplicar cualquiera de las interpolaciones mencionadas anteriormente en este capítulo, se corre el riesgo de provocar un desenfoque, ese desenfoque puede solventarse, en gran medida, utilizando las técnicas de sharpening o afilado, o, por supuesto, aplicando las redes neuronales entrenadas en esta

investigación para tal efecto; todo esto viene recogido más adelante en este capítulo. El proceso de sharpening es muy valioso a la hora de reparar el enfoque de una imagen pequeña, hecha más grande a través de la interpolación; sin embargo, su eficacia está limitada a niveles bajos de desenfoque; no ocurre lo mismo con los modelos creados, los cuales son capaces de generar imágenes enfocadas mejor de lo que lo haría la propia imaginación humana, reconociendo patrones y detalles que escapan a nuestra percepción.

4.2. Aplicación de Redes Generativas Antagónicas de súper resolución o SRGAN

La superresolución de imagen se puede definir como el aumento del tamaño de las imágenes pequeñas, mientras se mantiene la caída de la calidad al mínimo, o la restauración de imágenes de alta resolución a partir de detalles obtenidos de imágenes de baja resolución. Este problema es bastante complejo, ya que existen múltiples soluciones para una imagen de baja resolución dada.

El aumento de resolución en imágenes tiene numerosas aplicaciones, de las cuales en este capítulo se profundizará en su aplicación sobre ortoimágenes satelitales y aéreas. Cabe mencionar que, hasta la fecha, este tipo de tecnología aún no ha sido aplicada para mejorar la resolución de las ortoimágenes aéreas, por lo que la propia aplicación para este propósito puede suponer una valiosa innovación, que puede beneficiar enormemente al desarrollo de la cartografía.

En este epígrafe se tratarán las redes del tipo SRGAN, que sin ser las únicas redes profundas de súper resolución, si son las que ofrecen resultados más prometedores y probablemente aplicables en el trabajo cartográfico; por esta razón, en las redes elaboradas se ha escogido una arquitectura similar a estas, concretamente cGAN o redes generativas antagónicas condicionales.

Como ya se ha visto en este capítulo, existen varias formas para aumentar la resolución de una imagen, algunas de las cuales son rápidas y precisas; sin embargo, todavía queda una cuestión por resolver: la recuperación de detalles de textura más finos de una imagen de baja resolución, evitando que la imagen se distorsione.

Los trabajos más recientes con redes neuronales se han centrado en gran medida en minimizar el error de reconstrucción cuadrático medio; no obstante, los resultados obtenidos muestran un alto nivel de generación de ruido, lo que significa que tenemos buenos resultados de calidad de imagen, pero a menudo carecen de detalles y son perceptiblemente insatisfactorios, ya que no pueden igualar la fidelidad esperada en imágenes de alta resolución. Como se ha visto en epígrafes anteriores, los algoritmos tradicionales de interpolación intentan ver similitudes en el espacio de píxeles, lo cual conduce a resultados poco satisfactorios o producen imágenes borrosas, esto ocurre también con muchas de las redes neuronales desarrolladas hasta la fecha.

En cuanto a la arquitectura de las SRGAN, este tipo de red generativa antagónica aplica una red profunda en combinación con una red adversaria para producir imágenes de mayor resolución. Para el entrenamiento realizado en las SRGAN, se procesan las imágenes de alta resolución para obtener imágenes de baja resolución; de esta manera se obtienen imágenes de alta y baja resolución para usar como datos de entrenamiento. La metodología de entrenamiento utilizada en las SRGAN es prácticamente idéntica a la aplicada para entrenar los modelos generados en esta tesis para el aumento de resolución de ortoimágenes aéreas.

Una de las diferencias de la SRGAN con respecto a las redes aquí desarrolladas es que la SRGAN utiliza el modelo preentrenado VGG19 para mejorar los resultados; esto ofrece buenos resultados en la mayoría de las fotografías. Sin embargo, en lo que al trabajo cartográfico se refiere, supone un inconveniente, ya que el modelo VGG19 no ha sido entrenado con ortoimágenes aéreas ni satelitales, por lo que empeorara la calidad de los resultados en el proceso de mejora de la resolución cuando es aplicado a la cartografía.

4.3. Comparativa entre los resultados de modelos de Deep Learning existentes y del modelo de creación propia DeepMapsDespixela

A pesar de que los actuales modelos de redes neuronales existentes permiten una mejora de la resolución, que supera ampliamente los resultados obtenidos con los métodos de interpolación tradicionales, al ser aplicados a datos cartográficos (la aplicación de modelos de redes neuronales para la mejora de la resolución de ortoimágenes se ha realizado por primera vez en esta tesis) sus resultados son poco satisfactorios en comparación con los modelos que han sido expresamente entrenados para el tratamiento de este tipo de datos. A continuación, se puede comprobar como los resultados ofrecidos por el modelo DeepMapsDespixela, al ser aplicado sobre ortoimágenes aéreas, son muy superiores a los obtenidos con cualquiera de las otras técnicas existentes.

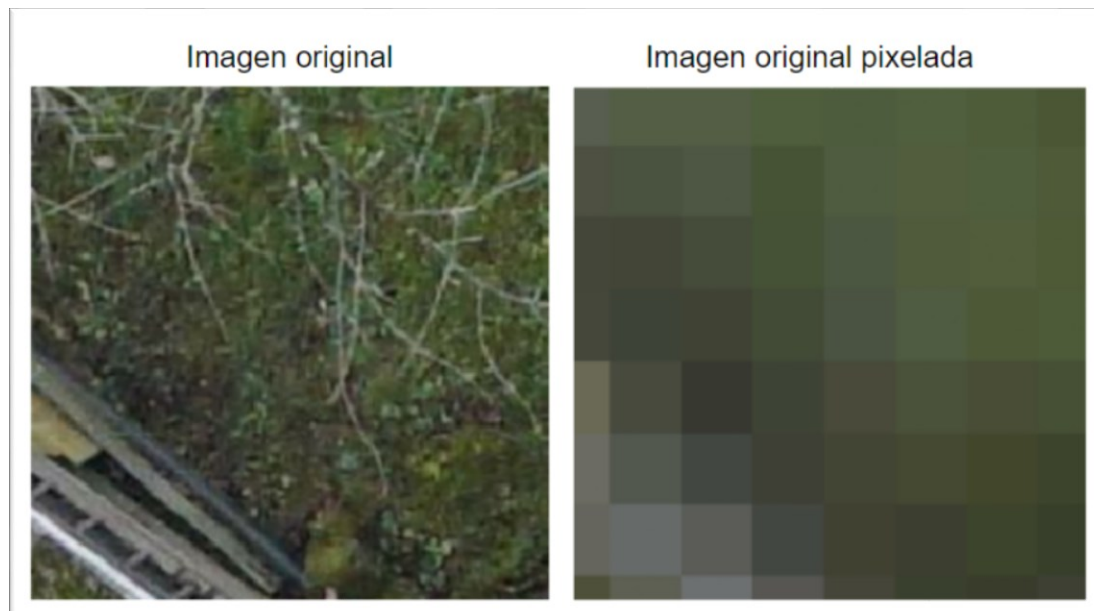


Figura 4.8. Pixelación de ortoimagen para las pruebas comparativas
Fuente: elaboración propia, a partir de datos de Wingtra

4.3.1. Resultados obtenidos comparados con la imagen original

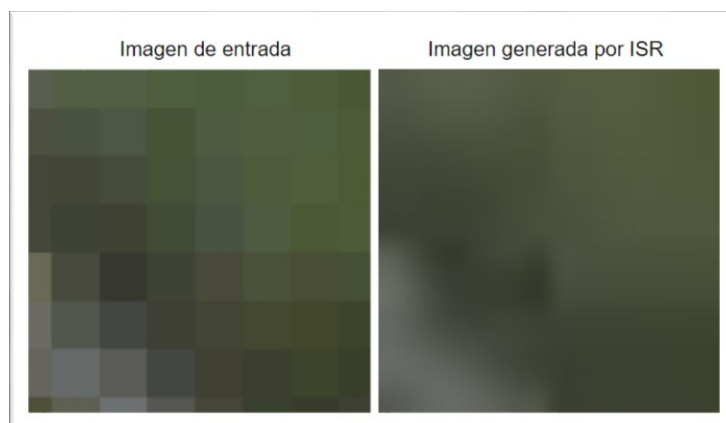


Figura 4.9. Modelo Image Super Resolution (ISR) aplicado a una ortoimagen en color
Fuente: elaboración propia, a partir de datos de Wingtra



Figura 4.10. Modelo Image Super Resolution + Modelo Noise Cancel aplicado a una ortoimagen en color
Fuente: elaboración propia, a partir de datos de Wingtra



Figura 4.11. Modelo Propio DeepMapsDespoxela aplicada a una ortoimagen en color
Fuente: elaboración propia a partir de datos de Wingtra

4.4. Técnicas clásicas para mejorar el enfoque

En este epígrafe se analizan distintos algoritmos existentes de mejora del enfoque, los cuales, a pesar de que resultan útiles en casos de desenfoque ligero, no pueden corregir problemas graves de desenfoque, a diferencia de lo que ocurre con los modelos desarrollados en esta investigación y que se exponen en el siguiente epígrafe.

4.4.1. Técnica de sharpening o afilado

Esta técnica resulta muy útil cuando se quiere mejorar los bordes de una imagen que no es nítida. A continuación, se muestran algunas imágenes que ejemplifican la eficacia de este proceso sobre ortoimágenes aéreas con un nivel medio de desenfoque.

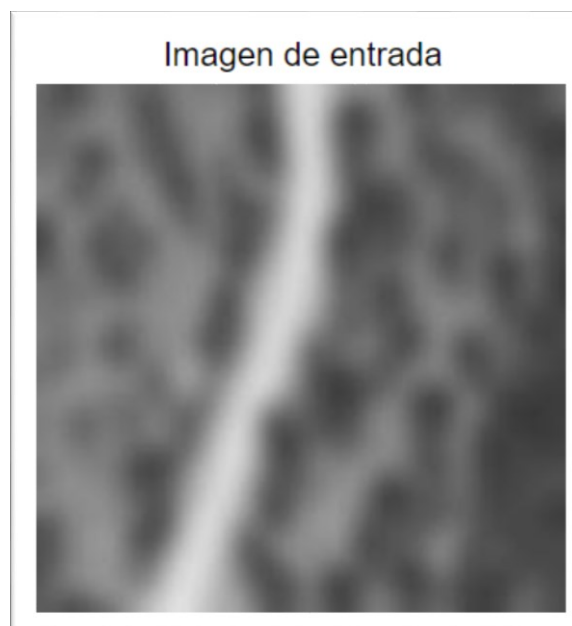


Figura 4.12. Ortoimagen desenfocada utilizada como entrada sobre la que se aplican los distintos kernels para las operaciones de sharpening mostradas en las siguientes figuras
Fuente: elaboración propia, a partir de datos del IGN

En cada una de las siguientes imágenes se ha aplicado un kernel o núcleo diferente para su corrección mediante un script de Python creado a tal efecto. Las distintas matrices de los núcleos o kernels utilizados, expresados como arrays de Numpy, son las siguientes:

- `kernel0 = np.array([[0, -1, 0],[-1, 5, -1],[0, -1, 0]], dtype="int")`
- `kernel1 = np.array ([[-1, -1, -1], [-1,9, -1], [-1, -1, -1]])`
- `kernel2 = np.array ([[1,1,1], [1, -7,1], [1,1,1]])`
- `kernel3 = np.array ([[-1, -1, -1, -1, -1], [-1,2,2,2, -1], [-1,2,8,2, -1], [-1,2,2,2, -1], [-1, -1, -1, -1, -1]]) / 8.0`

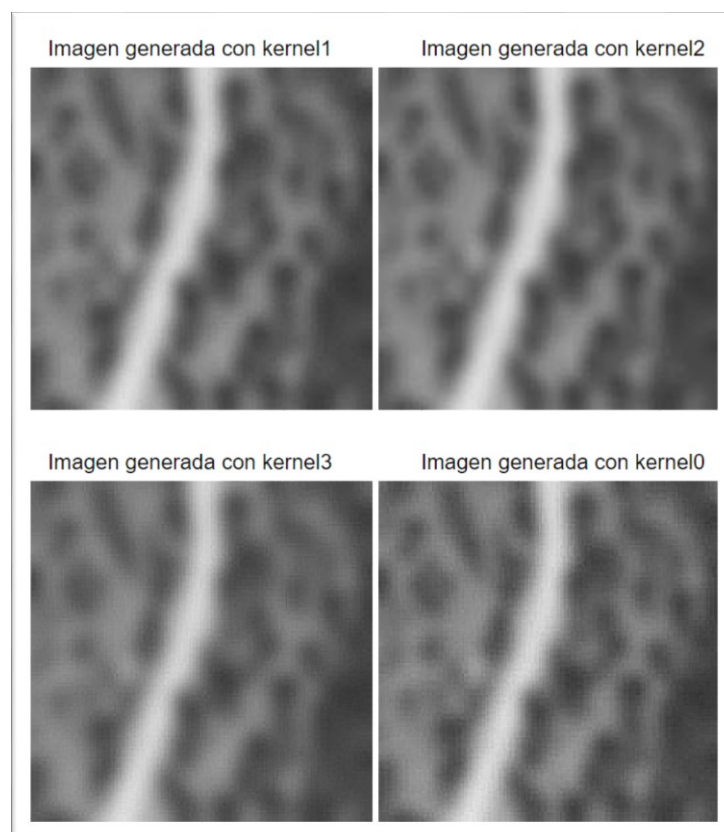


Figura 4.13. Resultados de la técnica de Sharpening con distintas configuraciones
Fuente: elaboración propia, a partir de datos del IGN

Como se puede observar en la figura anterior, el nivel de nitidez depende del tipo de kernel que usemos. En Python, usando la librería Numpy, se tiene mucha libertad para personalizar el kernel, y cada kernel proporciona un tipo diferente de afilado. Para enfocar una imagen, como lo estamos haciendo en la imagen generada con el kernel al que se ha dado el nombre de kernel1, usamos el siguiente núcleo:

$$M = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Pero el problema con esta técnica es que la imagen de salida parece mejorada artificialmente y no llega a eliminarse el desenfoque.

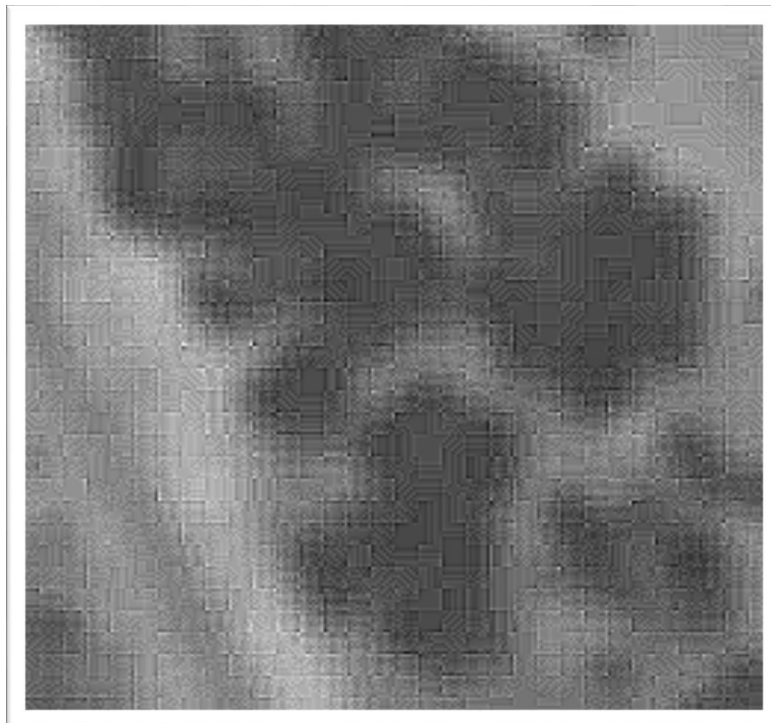


Figura 4.14. Aplicación extrema de sharpening
Fuente: elaboración propia, a partir de datos del IGN

Se puede intentar forzar el afilado mediante esta técnica, pero, aun así, no se consigue corregir el desenfoque cuando éste es demasiado acusado.

4.4.2. Enmascaramiento no nítido o Unsharp Masking

El enmascaramiento no nítido es una técnica utilizada para agudizar las imágenes, en la que se sustrae una versión borrosa de una imagen generada a partir de la imagen original. La fórmula de mezcla típica utilizada para el enmascaramiento no nítido es la siguiente: $\text{nítido} = \text{original} + (\text{original} - \text{borroso}) \times \text{cantidad}$. La cantidad es un parámetro que se puede ajustar al igual que el radio de desenfoque gaussiano utilizado.

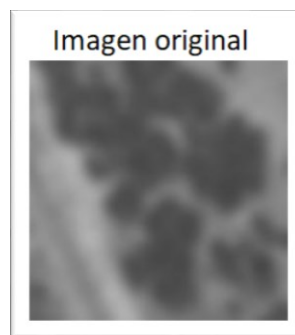


Figura 4.15. Ortoimagen desenfocada utilizada como entrada sobre la que se aplican distintas configuraciones para la aplicación de la técnica de Unsharp Masking mostradas en las siguientes figuras

Fuente: elaboración propia, a partir de datos del IGN

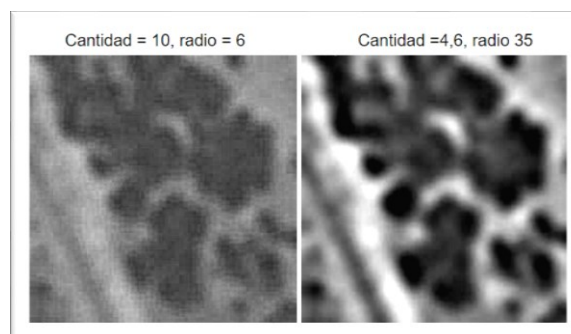


Figura 4.16. Resultados de la técnica de Unsharp Masking con distintas configuraciones

Fuente: elaboración propia, a partir de datos del IGN

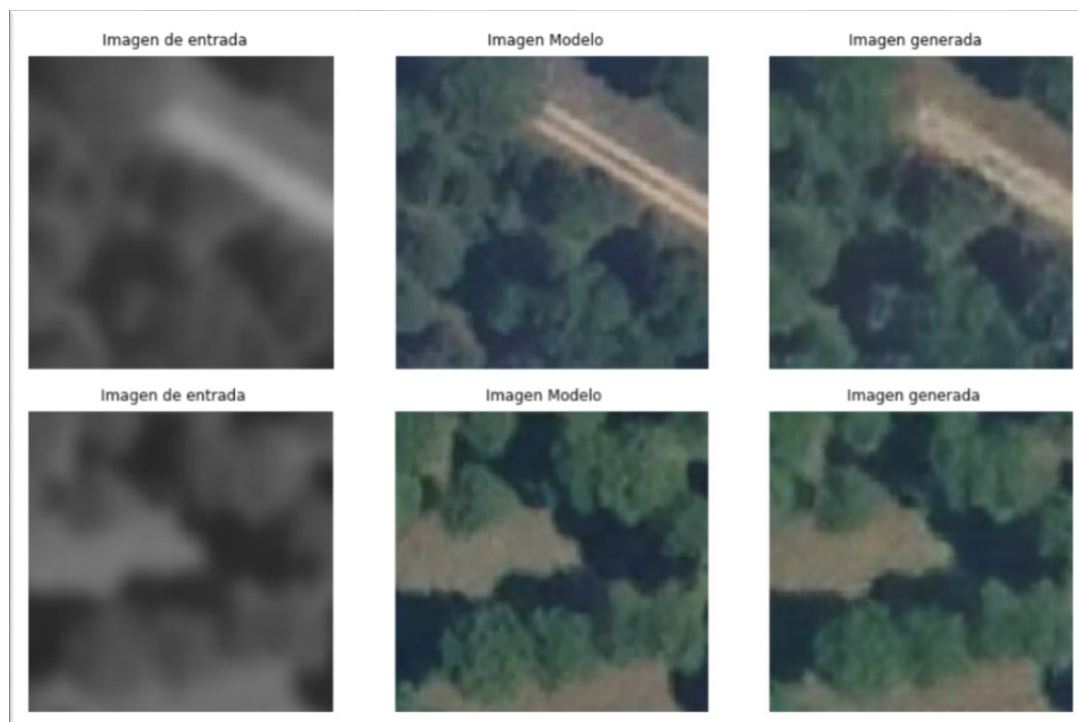
Como ya se ha dicho, podemos desenfocar una imagen para luego computar la imagen de detalle como la diferencia entre la imagen original y la desenfocada para implementar un enmascaramiento no nítido. La imagen nítida puede ser computada como una combinación lineal de la imagen original y la imagen de detalle.

4.5. Modelos propios

Las distintas inteligencias artificiales mencionadas a continuación han sido desarrolladas durante esta investigación y suponen las primeras entrenadas para solventar problemas de resolución y enfoque en datos cartográficos. Todas ellas tienen una arquitectura de redes generativas antagónicas condicionales, con un generador de tipo U-net y un discriminador de tipo PatchGAN, más conocidas como image-to-image o Pix2Pix; la arquitectura de este modelo viene descrita con mayor detalle en el capítulo 2 de este documento.

4.5.1. DeepMapsOptimiza

Este modelo ha sido entrenado con teselas de 256x256 píxeles, obtenidas de las hojas del PNOA correspondientes a la Comunidad de Madrid, con un nivel de zoom de 20 (escala aproximada 1:500), de las que se han seleccionado, de forma aleatoria, 3.512 imágenes durante 15.000 iteraciones.



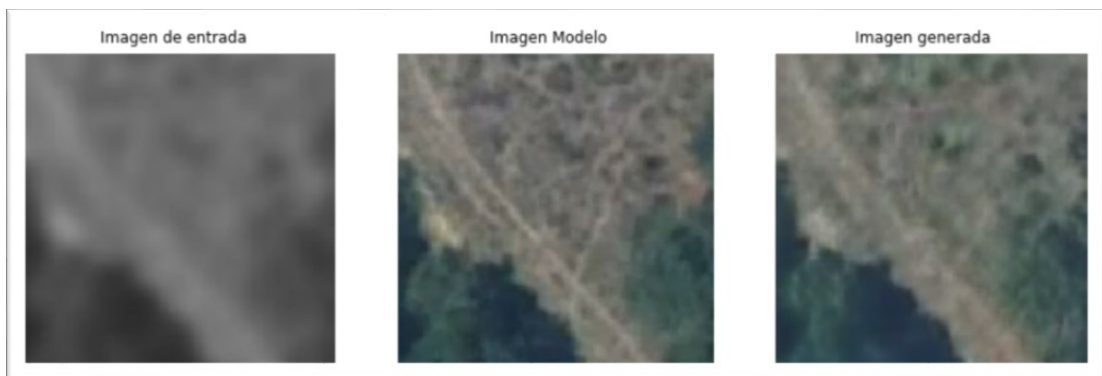


Figura 4.17. Resultados producidos por el modelo DeepMapsOptimiza en relación con las imágenes originales (Imagen Modelo)
Fuente: elaboración propia, a partir de datos del IGN

A estos datos se les realizó un tratamiento masivo para su ingesta en el modelo, generando imágenes a las que se les había quitado el color mediante la conversión de estas al espacio de color $L^*a^*b^*$ y extrayendo el canal L ; tras esto se generaron distintas imágenes de entrenamiento, añadiendo un desenfoque gaussiano a las mismas; el desenfoque gaussiano aplicado ha sido de 6 píxeles de radio.



Figura 4.18. Ejemplo de resultados obtenidos con el modelo DeepMapsOptimiza
Fuente: elaboración propia, a partir de datos del IGN

Como ocurre con las otras redes que desempeñan varias tareas, en este caso se ha considerado, como propiedad más significativa de esta red, su capacidad de enfocar y por lo tanto su menor especificidad en el dominio al que es aplicable.

4.5.2. DeepMapsEnfocayColoreaHR

Para el entrenamiento de esta red se ha recurrido a imágenes procedentes de drones, debido a que con esta prueba se pretende comprobar la posibilidad de aumentar la resolución, corregir el enfoque y añadir color a ortoimágenes aéreas, para lo cual era necesario partir de datos originales que contuviesen la información de los detalles que se querían obtener tras el procesamiento; de esta forma, para su entrenamiento se parte de imágenes de alta resolución como la mostrada a continuación.



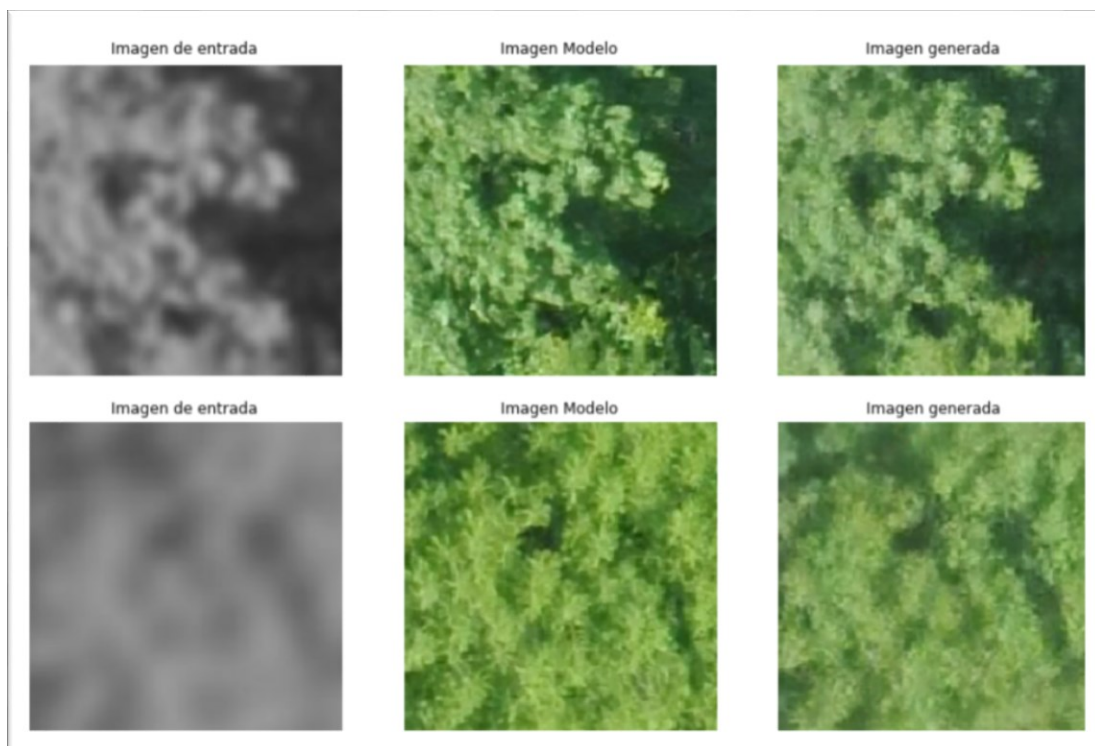
Figura 4.19. Ejemplo de imagen de alta resolución utilizada para el entrenamiento del modelo DeepMapsEnfocayColoreaHR

Fuente: elaboración propia, a partir de datos de Wingtra

Estas imágenes han sido captadas con una cámara Sony RX1RII, la cobertura del vuelo ha sido de 9 hectáreas y la resolución obtenida es de 1 centímetro por pixel. La captura de imágenes fue realizada en Zúrich (Suiza) y en Phoenix (EEUU) por la empresa Wingtra. Para el entrenamiento se han utilizado un total de 650 imágenes durante 3.200 iteraciones.

Al igual que en el caso del modelo anterior, las imágenes necesarias para el entrenamiento han sido generadas mediante un script de Python, a través del cual se ha eliminado el color mediante la extracción del canal L y se ha añadido un desenfoque gaussiano de distinta intensidad con el propósito de comprobar que una misma red podría resolver problemas de desenfoque, independientemente de la intensidad de este; los radios de desenfoque gaussiano aplicados son de 2, 4, 6 y 8 píxeles. Todas estas operaciones de tratamiento de imagen se realizaron de forma masiva mediante un script de Python creado para tal fin, el cual, además, creaba los pares de imágenes necesarios para el entrenamiento.

A continuación, se muestran algunos de los resultados obtenidos al procesar imágenes con distinto nivel de desenfoque con el modelo DeepMapsEnfocayColoreaHR.



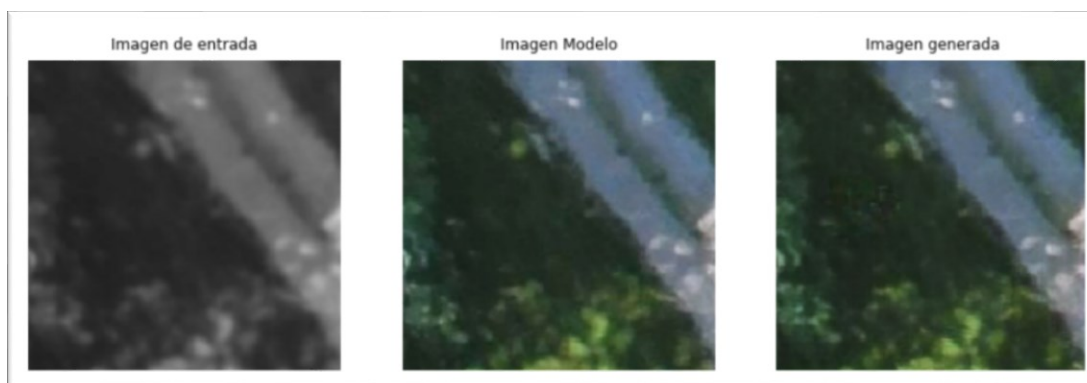


Figura 4.20. Resultados producidos por el modelo DeepMapsEnfocayColoreaHR en relación con las imágenes originales (Imagen Modelo)

Fuente: elaboración propia a partir de datos de Wingtra

Este modelo se ha creado con el propósito doble de colorear las imágenes en blanco y negro y corregir su enfoque. Aunque este tipo de inteligencia artificial sería posible encuadrarla tanto en el capítulo 3 como en el capítulo 4, los detalles de esta vienen recogidos en el capítulo 4, ya que, para la coloración de imágenes, en esta tesis se han desarrollado otras inteligencias artificiales que devuelven mejores resultados; sin embargo, ésta tiene la ventaja de que, gracias al desenfoque admitido de las imágenes de origen, resulta más flexible en la aplicación sobre imágenes de diferentes dominios.

4.5.3. DeepMapsOptimiza17BN

Este modelo ha sido entrenado para corregir el desenfoque en ortoimágenes en escala de grises, pero sin añadir color a la ortoimagen original. Como datos de entrenamiento se han utilizado pares de imágenes generados a partir de teselas extraídas de las hojas del PNOA de Máxima Actualidad, correspondientes a la Comunidad de Madrid. Las teselas generadas son de 256x256 píxeles, con un nivel de zoom de 17, equivalente a una escala aproximada de 1:4.000, a partir de las teselas originales se han producido los pares de imágenes (imagen de entrada e imagen modelo) sometiéndolas a un

tratamiento para eliminar el color, mediante la extracción del canal L del espacio de color $L^*a^*b^*$; además a las ortoimágenes tomadas como imágenes de entrada se les ha aplicado un filtro de desenfoque gaussiano. Con las ortoimágenes a las que se les ha modificado el enfoque y las originales (imágenes modelo), ambas en escala de grises, se han creado los pares de imágenes necesarios.

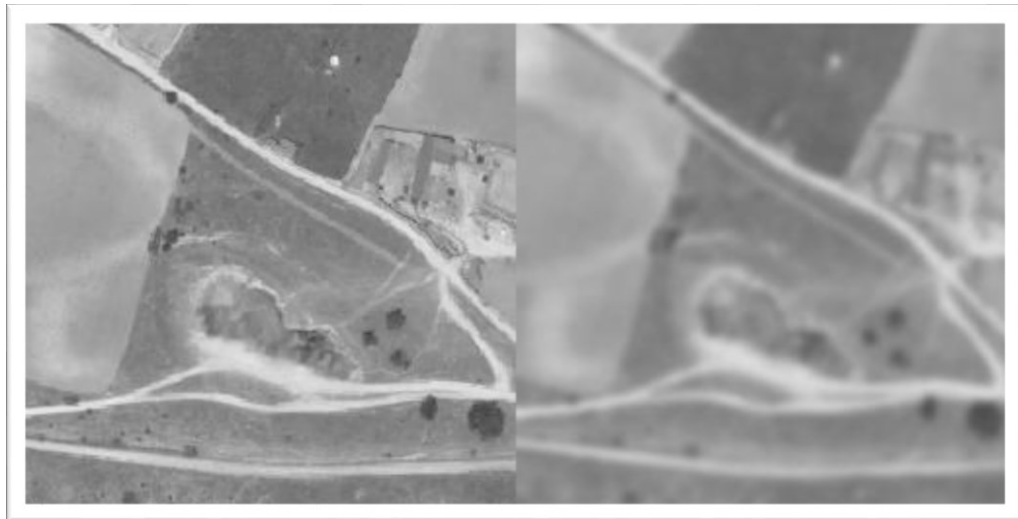
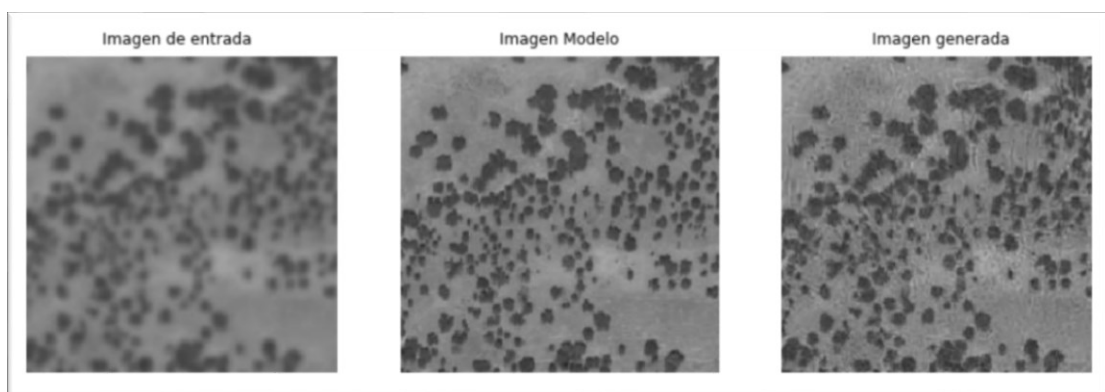


Figura 4.21. Ejemplo de par de imágenes generado para el entrenamiento del modelo DeepMapsOptimiza17BN

Fuente: elaboración propia, a partir de los datos del IGN

El total de imágenes con las que se ha entrenado el modelo es de 3.852 imágenes y se han realizado 6.500 iteraciones.



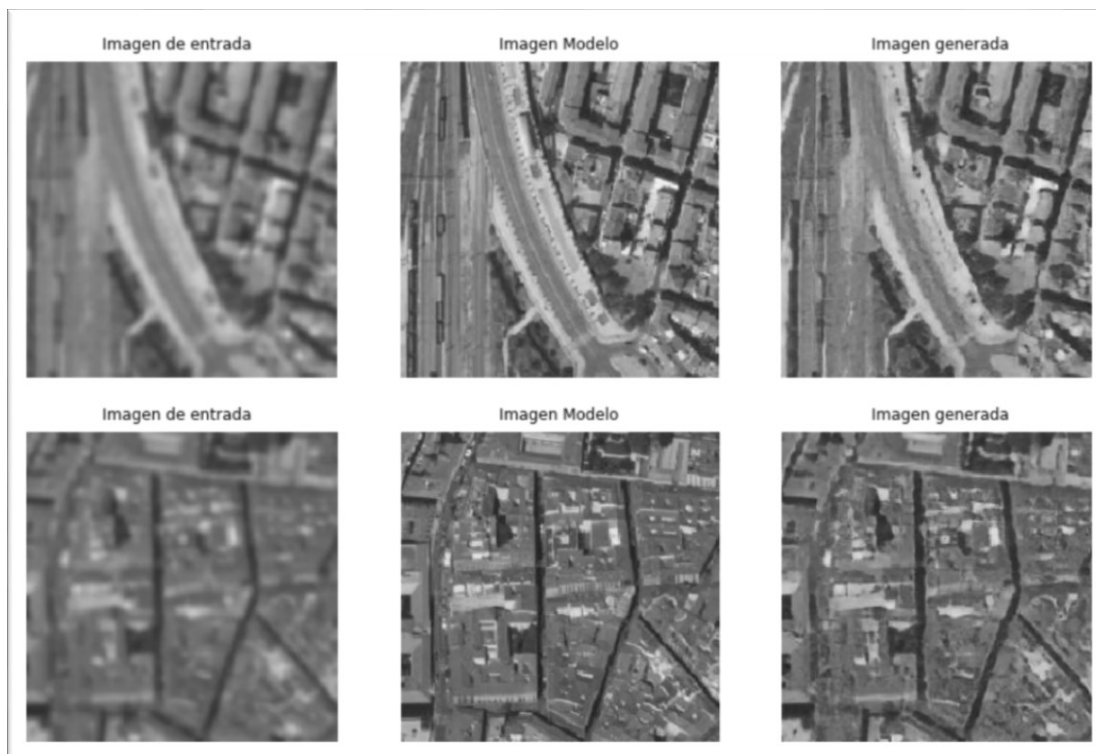


Figura 4.22. Resultados producidos por el modelo DeepMapsOptimiza17BN en relación con las imágenes originales (Imagen Modelo)

Fuente: elaboración propia, a partir de los datos del IGN

Al igual que el modelo anterior, en este caso la red cumple la función de enfocar las ortoimágenes aéreas, en este caso en blanco y negro, dando mejores resultados al trabajar con un nivel de zoom de 17 o, lo que es lo mismo, con una escala aproximada de 1:4.000, ya que es con este nivel con el que ha sido entrenada la red.



Figura 4.23. Ejemplo de resultados obtenidos con el modelo DeepMapsOptimiza17BN

Fuente: elaboración propia, a partir de los datos del IGN

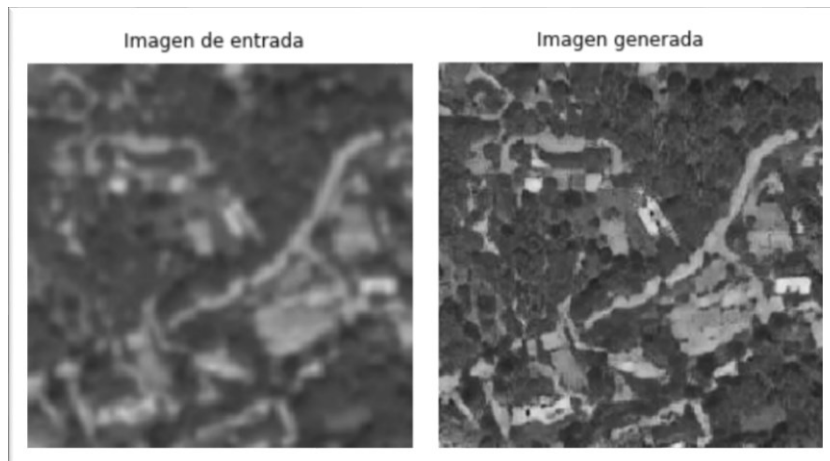


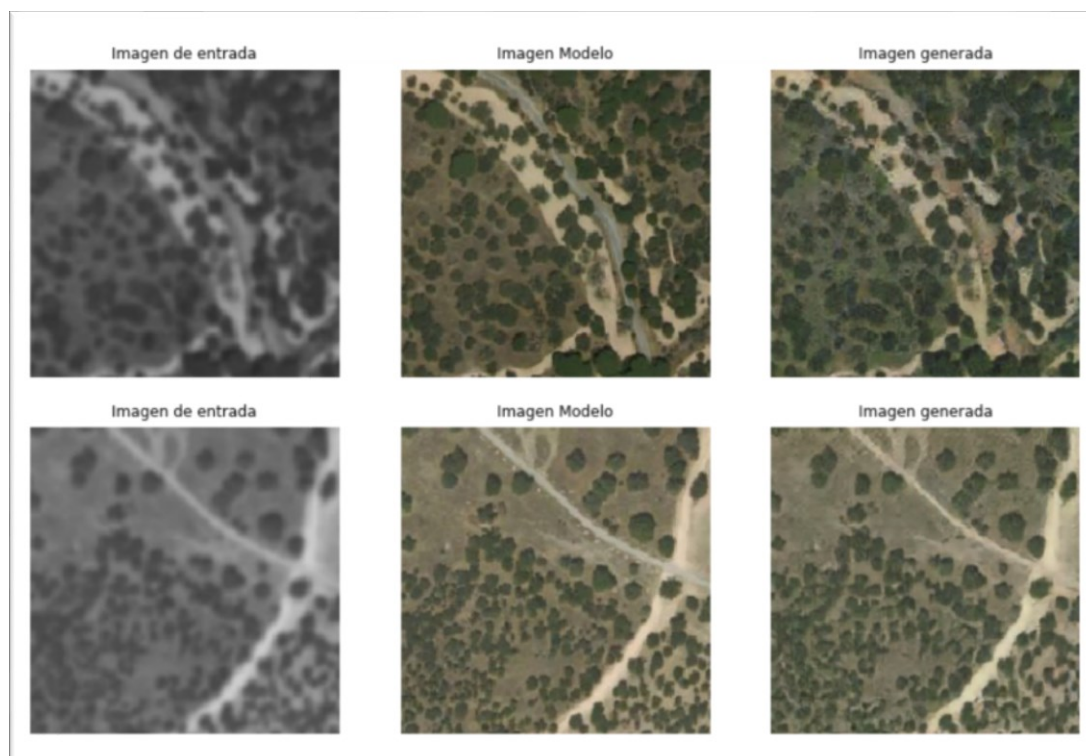
Figura 4.24. Ejemplo de resultados obtenidos con el modelo DeepMapsOptimiza17BN
Fuente: elaboración propia, a partir de los datos del IGN

En este caso se ha optado por limitar las funciones sólo al enfoque, ya que, por los resultados obtenidos, se ha comprobado que las inteligencias artificiales desarrolladas son más eficaces al tener que realizar una sola tarea. Debido a esto, para obtener resultados de la máxima calidad es preferible procesar una ortoimagen en escala de grises desenfocada, en primer lugar, para su enfoque y, tras ello, procesarla para su coloración. En este tipo de casos debe valorarse la estrategia a seguir en función de lo prioritario, la velocidad o la calidad, siendo ésta, en cualquier caso, superior a la obtenida con cualquiera de los sistemas existentes en el momento en el que se ha elaborado este documento. Además, cabe destacar la gran utilidad de esta red para realizar labores de conversión de dominio, pudiendo actuar como puente entre ortoimágenes de distintos orígenes y las inteligencias desarrolladas entrenadas con datos pertenecientes a dominios diferentes a los datos a tratar.

4.5.4. DeepMapsOptimiza17

Este modelo es similar al anterior, con la salvedad de que, además de ser entrenado para la corrección del enfoque en ortoimágenes en escala de grises, también ha sido entrenado para la coloración de estas. El dataset escogido para el entrenamiento de este modelo es el mismo que para el modelo anterior, constando de 3.852 imágenes con las que se han realizado 6.500 iteraciones. La diferencia con el modelo anterior es que las ortoimágenes tomadas como imágenes modelo no han sido transformadas a escala de grises, permitiendo, de esta manera, que la inteligencia artificial DeepMapsOptimiza17 aprenda cómo asignar color además de corregir el desenfoque.

Al igual que en el modelo anterior las teselas utilizadas para la creación de los pares de imágenes de entrenamiento son de 256x256 píxeles, con un nivel de zoom de 17 equivalente a una escala aproximada de 1:4.000.



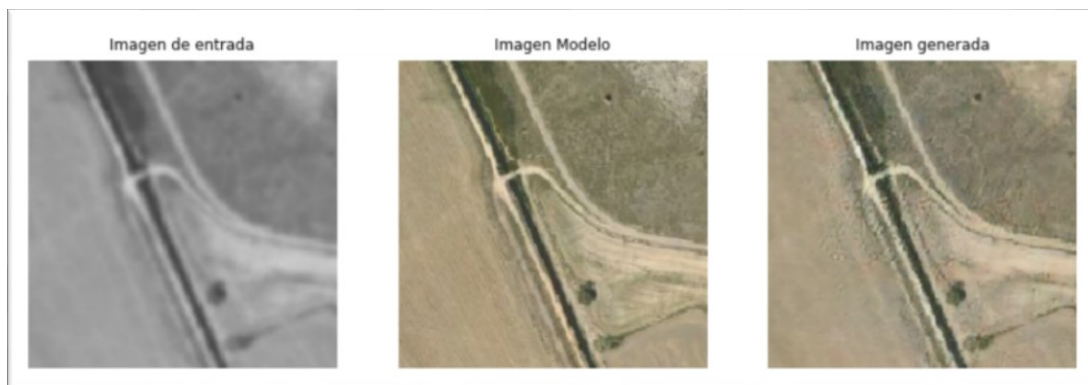


Figura 4.25. Resultados producidos por el modelo DeepMapsOptimiza17 en relación con las imágenes originales (Imagen Modelo)
Fuente: elaboración propia, a partir de los datos del IGN

Como se comprueba con los resultados mostrados anteriormente, esta inteligencia artificial es capaz de corregir el enfoque y asignar color de forma verosímil, siendo su escala de trabajo más óptima 1:4.000.

4.5.5. DeepMapsDespixela

Este modelo ha sido entrenado para mejorar la resolución de imágenes pixeladas, concretamente ha sido entrenado con el mismo dataset que el modelo DeepMapsEnfocayColoreaHR (datos de Zúrich (Suiza) y Phoenix (EEUU) suministrados por la empresa Wingtra), descrito en el subepígrafe 4.5.2. En este caso, las teselas de 256x256 han sido tratadas con script de Python creado expresamente para darles una resolución de 16x16 píxeles y entrenar al modelo para que fuese capaz de recrear una imagen lo más fiel posible a la original a partir de la imagen pixelada. A continuación, se muestra un ejemplo de los pares de imágenes generados para el entrenamiento de este modelo.



Figura 4.26. Ejemplo de par de imágenes de entrenamiento generado para el entrenamiento del modelo DeepMapsDespixela
Fuente: elaboración propia, a partir de datos Wingtra

Este modelo, a diferencia del modelo DeepMapsEnfocayColoreaHR ha sido entrenado con una muestra de 2.746 imágenes durante 2.000 iteraciones. Los resultados, aunque muy satisfactorios y superiores a cualquier tecnología existente en la actualidad, podrían mejorarse aumentando el número de iteraciones del entrenamiento.



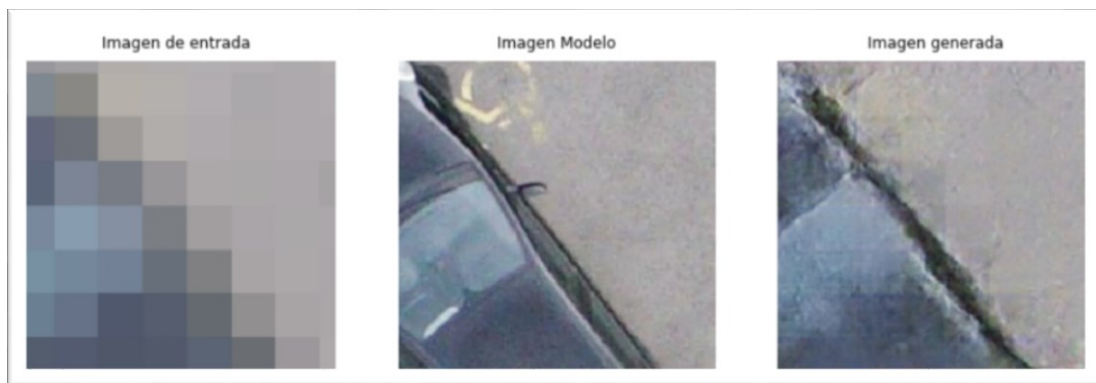


Figura 4.27. Resultados producidos por el modelo DeepMapsDespoxela en relación con las imágenes originales (Imagen Modelo)
Fuente: elaboración propia, a partir de datos Wingtra

El modelo DeepMapsDespoxela permite generar una ortoimagen de alta resolución a partir de una ortoimagen pixelada. Los resultados obtenidos son muy superiores a los que se consiguen mediante interpolación o aplicando redes de Deep Learning de carácter más generalista, ya que, como anteriormente se ha indicado, las presentadas en esta tesis son las primeras inteligencias artificiales entrenadas específicamente para la reparación de la pixelación en ortoimágenes aéreas. Los datos inexistentes, al generar una nueva ortoimagen de alta resolución los crea de la forma estadísticamente más verosímil, en base a los datos con los que la red ha sido entrenada y los recibidos en la imagen de entrada, originando un resultado que va a ser realista, pero va a estar constreñido a imágenes detalladas pero concordantes con información disponible en la imagen de entrada.



Figura 4.28. Ejemplo de resultados obtenidos con el modelo DeepMapsDespoxela
Fuente: elaboración propia, a partir de datos Wingtra

5. Corrección de errores y artefactos

Uno de los problemas habituales en el trabajo cartográfico es el hecho de toparse con errores o artefactos, ya sean producidos durante la propia captura de datos o durante los procesos de edición cartográfica; a veces, incluso son intencionados, debidos a censuras, ya que se eliminan complejos militares y zonas sensibles. Hasta ahora, la metodología de trabajo habitual, en caso de querer eliminarlos, era la edición manual de imágenes con software especializado para el tratamiento fotográfico, como por ejemplo Adobe Photoshop, pero, debido al alto coste de recursos temporales y humanos que requiere, la tarea se restringía sólo a casos muy concretos (y los resultados generalmente no eran lo suficientemente verosímiles como para engañar a un humano). En este capítulo vamos a analizar la posibilidad de automatización de esta laboriosa tarea mediante el uso de algoritmos ya habituales en el tratamiento de imágenes y la posibilidad de aplicar el Deep Learning a la detección y corrección automatizada de estos problemas, presentes en muchas cartografías.

En primer lugar, por supuesto, existe el reto de identificarlos, para posteriormente proceder a su corrección. Cabe mencionar que no todos los errores y artefactos son iguales, por lo que las técnicas utilizadas tradicionalmente para solucionar estos problemas no eran los mismos en todos los casos.

En las cartografías más antiguas existen problemas derivados de la propia conservación del soporte físico de las mismas, pueden tener araños, manchas, o, incluso, partículas de polvo presentes en el momento de su digitalización; este tipo de problemas, cuando no revisten demasiada importancia, en base a las pruebas realizadas, normalmente pueden ser resueltos mediante interpolaciones o técnicas basadas en ecuaciones diferenciales parciales; sin embargo, en los casos en los cuales los errores son

de mayores dimensiones, este tipo de técnicas resultan ineficaces, ya que en la propia imagen a tratar no se dispone de información suficiente como para completar los datos faltantes. Es por ello por lo que debe recurrirse a las inteligencias artificiales, las cuales son capaces de generar unos datos coherentes para completar la imagen, no basándose únicamente en los datos de la imagen a tratar, sino en todo el conocimiento adquirido durante su entrenamiento con miles de imágenes.



Figura 5.1. Ejemplo de aplicación del modelo propio DeepMapsInpainting
Fuente: elaboración propia, a partir de los datos del IGN

Gracias a esta capacidad de las redes neuronales profundas, se han podido crear durante esta investigación las primeras inteligencias artificiales capaces de corregir errores o completar la información ausente, incluso en áreas extensas, de las ortoimágenes aéreas, tal y como se muestra, al final de este capítulo, en los resultados de los modelos de creación propia. De todas formas, aunque la corrección mediante redes generativas antagónicas puede suponer un avance en la cartografía, su consecución también ha requerido un entrenamiento bastante costoso en términos computacionales. Por ello, es conveniente analizar los algoritmos y modelos previos a esta tesis para entender el problema y la posible necesidad de recurrir a estas tecnologías, puesto que, según el caso, puede bastar con aplicar tecnologías ya presentes en otros ámbitos, como la edición fotográfica, al mundo de la cartografía digital. Esto es necesario tenerlo en cuenta, ya que no todos los métodos de corrección tienen un mismo coste computacional ni de tiempo, por lo que cuanto más sencillo sea

el algoritmo utilizado, menos tiempo consumirá, razón por la cual no tiene sentido aplicar una tecnología más avanzada a la resolución de problemas que podrían solventarse de una forma más sencilla y económica. A este respecto, cabe mencionar que los problemas de errores y ruido presentes en la toma de datos no sólo afectan a las ortoimágenes aéreas o satelitales, sino también a las capturas de datos LiDAR, pero el tratamiento de las mismas tiene sus propias particularidades; es por ello que, para solventar este tipo de inconvenientes presentes en las nubes de puntos, se ha desarrollado un software específico para la corrección y mejora de datos LiDAR (MEL), con el cual se pueden no sólo eliminar las capturas de datos erróneas, sino que también se puede completar la información ausente en algunos puntos y aumentar su resolución.

Para el tratamiento de los datos LiDAR, a diferencia de lo que ocurre con los datos de ortoimágenes, se ha optado por no aplicar redes neuronales, ya que, debido al volumen y complejidad de las nubes de puntos en tres dimensiones, los costes computacionales serían excesivos. Por este motivo, para la corrección de estos datos se ha recurrido a la aplicación de algoritmos más clásicos. Los detalles, tanto del desarrollo como de la utilización del software de tratamiento y visualización de datos LiDAR, se encuentran en los capítulos 8 y 9, en los cuales también se exponen los resultados obtenidos, así como su aplicación en casos de uso reales.

5.1. Ruido

Uno de los errores más comunes en las ortoimágenes, tanto analógicas como digitales, es la presencia de ruido, el cual constituye la principal barrera para el procesamiento efectivo de la señal y, por lo tanto, del procesamiento de la imagen.

El ruido es la variación de la señal con respecto a su verdadero valor en una pequeña cantidad, de forma aleatoria, debido a factores externos o internos en la captura, transmisión y procesamiento de la imagen. Estos factores no pueden ser fácilmente controlados, y por lo tanto se introducen elementos aleatorios en los resultados del procesamiento, que pueden ser dependientes o independientes del contenido de la imagen. Debido a que el ruido es un problema fundamental en el procesamiento de imágenes, una gran parte de los algoritmos de procesamiento de estas, y por lo tanto el desarrollo de los algoritmos de interpolación de imágenes, se dedican a aumentar la robustez los algoritmos frente al ruido.

A pesar de que el ruido en las imágenes digitales puede originarse en una variedad de fuentes, algunas de ellas son debidas a la propia codificación de la imagen en formatos comprimidos como el jpg., en este caso se hace foco en el ruido de captura (concretamente el ruido de impulso conocido como “sal y pimienta”), ya que el resto de tipos de ruido son corregidos de forma automática con prácticamente todos los modelos generativos aquí desarrollados o con algoritmos ya existentes como los mostrados en el siguiente subepígrafe de forma bastante satisfactoria.

El ruido en las imágenes también puede ser caracterizado, a grandes rasgos, en dos clases principales: aditivo y multiplicativo. Un ejemplo de ruido multiplicativo es la iluminación variable, en la que el artefacto visual de la imagen es causado por una iluminación desigual (este tipo de ruido se ha corregido con

muy buenos resultados utilizando los modelos DeepMapsRestaura 15, 17 y 19, los cuales vienen explicados en los últimos subepígrafes de este capítulo). Se suele suponer que el ruido aditivo es un ruido de impulso (también conocido como ruido de "sal y pimienta"). El ruido de impulso altera al azar el valor de algunos píxeles. Como resultado, la imagen binaria corrompida por el ruido de impulso tendrá algunos píxeles que se vuelven blancos en el área de imagen negra y negros en el área de imagen blanca. Por eso, este ruido también se conoce como ruido "sal y pimienta". Este ruido es un problema muy común en la adquisición de imágenes digitales, donde la aleatoriedad aparece a partir de los elementos de hardware porque el comportamiento del dispositivo electrónico (en particular el dispositivo de captura de imágenes) puede ser aleatorio debido a las condiciones térmicas; dicho comportamiento es intrínsecamente un fenómeno de flujo de carga y depende en gran medida de la temperatura. Este tipo de ruido, cuando se da de forma intensa, es bastante difícil de corregir, razón por la que se han entrenado varios modelos para realizar esta tarea, obteniendo unos resultados muy satisfactorios, tal y como puede apreciarse en la siguiente imagen, donde se ha corregido el ruido mediante la aplicación del modelo DeepMapsRuido.



Figura 5.2. Ejemplo de aplicación del modelo propio DeepMapsRuido
Fuente: elaboración propia, a partir de los datos del IGN

Dentro de los algoritmos existentes en la actualidad para poder solventar el problema del ruido, deben destacarse los surgidos en los últimos años, ya que dan resultados mucho mejores que los anteriores. Dentro de éstos, los que mejores resultados devuelven son los basados en la eliminación del ruido mediante medios no locales y los llamados métodos de inpainting. Este es el motivo por el que este capítulo se centra en la revisión de los métodos actuales más prometedores y en los propios modelos de Inteligencia Artificial en esta investigación desarrollados. Tanto unos como otros suponen una novedad con respecto a los métodos anteriores, pues éstos se basaban, únicamente, en completar o corregir la información ausente o errónea promediando los valores de los píxeles colindantes. Cabe mencionar que la mayoría de los métodos conocidos como no locales, en realidad deberían llamarse métodos de vecindad ampliada, dado que la búsqueda de píxeles similares se hará en un vecindario más grande, pero aun localmente. Sin embargo, el término está ya establecido por el uso y por esa razón lo mantenemos en este documento.

5.1.1. Non-Local Means Denoising

Este método fue propuesto por Antoni Buades, Bartomeu Coll y Jean-Michel Morel; es uno de los más novedosos y que mejores resultados ha aportado en la eliminación de ruido sin recurrir a la inteligencia artificial; sin embargo, los resultados del mismo, aunque no tan satisfactorios como los métodos basados en redes neuronales convenientemente entrenadas, como las que se presentan en esta tesis, se obtienen con un coste menor de computación, por lo que su aplicación podría ser preferible en determinados supuestos en los que este método bastase para la corrección de las ortoimágenes aéreas.

El método se basa en un principio simple, reemplazar el color de un píxel con un promedio de los colores de similares píxeles. Pero los píxeles más

parecidos a un píxel dado no tienen por qué ser cercanos entre sí, esto es lo que marca la diferencia en los resultados con respecto a otros algoritmos anteriores que se basaban en promediar la información de los píxeles colindantes. Debido a la necesidad de este análisis no local, este algoritmo analiza una gran parte de la imagen en busca de todos los píxeles que se asemejan al píxel a corregir; como consecuencia, este proceso tiene un mayor coste de computación que otros algoritmos anteriores, sin embargo, sus resultados son mucho mejores. Este método, a diferencia de otros métodos no locales que se basan en la ampliación del vecindario de píxeles que se analizan para ser promediados, se fundamenta en una versión de parche de los métodos no lineales, es decir, se analizan varias porciones de la imagen para detectar aquellas porciones o parches similares a la que se quiere corregir y realizar el promedio de ésta en términos de una vecindad ampliada con respecto a los métodos más clásicos.

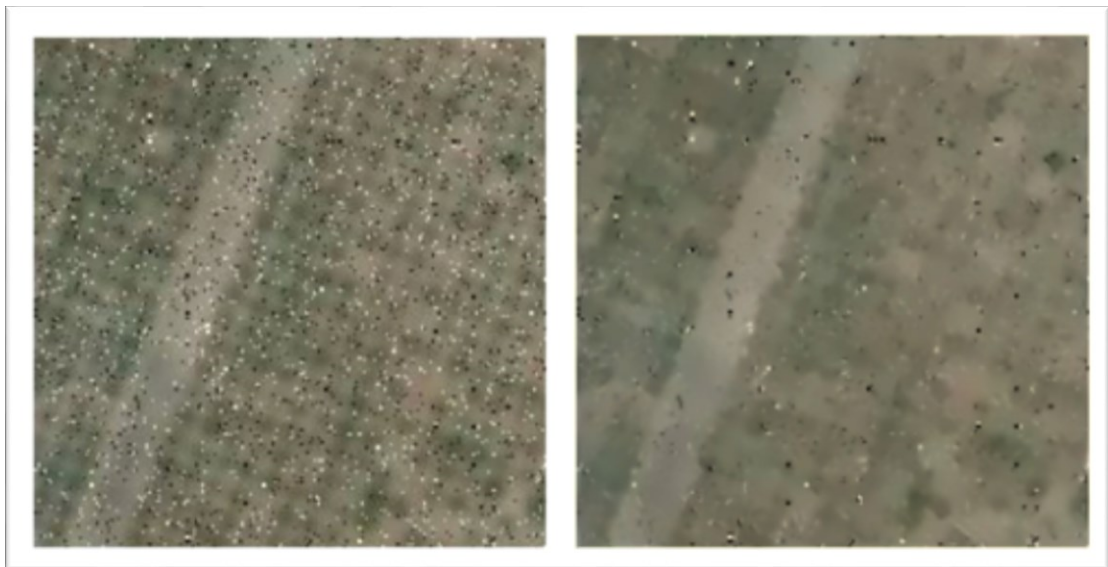


Figura 5.3. Ejemplo de aplicación del método Non-Local Means Denoising
Fuente: elaboración propia, a partir de los datos del IGN

Como puede apreciarse en la imagen superior, este método corrige el ruido en gran medida, aunque no completamente, pero aumenta el desenfoque ya existente en la imagen original. Sin embargo, para evitar esta consecuencia negativa del proceso de corrección del ruido, ha sido entrenada la inteligencia

artificial “DeepMapsRuidoyDesenfoque”, la cual es capaz de corregir el ruido y el desenfoque de forma simultánea y con unos resultados muy superiores a cualquier otro algoritmo o modelo de Deep Learning existente actualmente, al menos al ser aplicada en ortoimágenes aéreas, tal y como se muestra a continuación.

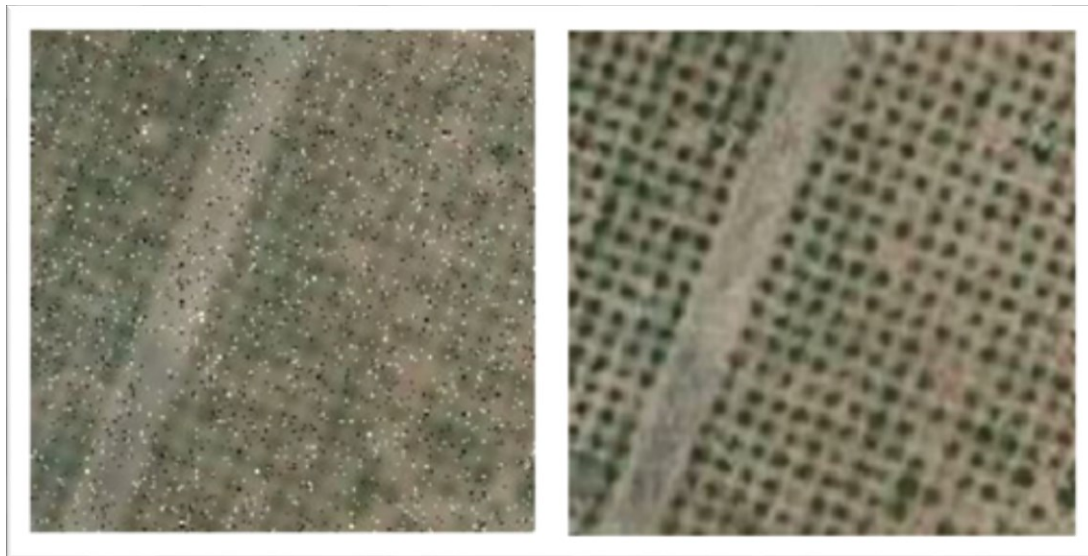


Figura 5.4. Ejemplo de aplicación del modelo propio DeepMapsRuidoyDesenfoque
Fuente: elaboración propia, a partir de los datos del IGN

El método “Non-Local Means Denoising” viene ya implementado en varias funciones incluidas dentro de la librería de Python OpenCV, por lo que su aplicación resulta muy sencilla, requiriendo además un coste computacional muy bajo, en comparación con el consumo de recursos del entrenamiento de las redes neuronales. Debido a lo anteriormente expuesto, pese a que sus resultados no alcanzan la calidad de los producidos por las inteligencias artificiales creadas en esta investigación, su aplicación en el tratamiento masivo de datos mediante un script de Python ha de valorarse.

5.2. Inpainting o relleno de la imagen

La técnica de Inpainting en sí no es una técnica nueva, hace ya siglos que viene aplicándose de forma manual; de hecho, proviene de la restauración de arte, donde también se le llama "retoque".

La necesidad de retocar la imagen de una manera discreta se extendió naturalmente de las pinturas a la fotografía y al cine. Los propósitos seguían siendo los mismos, revertir el deterioro, como el producido por los arañazos y las manchas de polvo en la película, o añadir o quitar elementos con la conocida como técnica del aerógrafo, muy utilizada en censuras. En el ámbito digital, el problema del Inpainting apareció por primera vez en las telecomunicaciones, donde se necesitaba rellenar los bloques de imagen que se habían perdido durante la transmisión de datos.

Como se ha mencionado anteriormente no todos los errores o artefactos son iguales; por ello, en este epígrafe vamos a analizar las posibilidades de corrección, con tecnologías ya existentes, de los artefactos producto de la falta de datos en distintas áreas, ya sea por errores en la captura, censura o deterioro de los medios. Dada la multitud de errores posibles en las fotografías, en los últimos años han adquirido especial relevancia los métodos de Inpainting, los cuales proporcionan un medio para la reconstrucción de pequeñas porciones dañadas de una imagen. De esta forma puede sustituirse la información dañada, sea cual sea la naturaleza del daño, por una reconstrucción de esa zona.

El funcionamiento de la mayoría de los métodos de Inpainting consiste, primeramente, en seleccionar las regiones de la imagen que se van a pintar, normalmente de forma manual, creando una máscara; hecho esto, la información de color se propaga hacia el interior de la región a completar, a partir de los límites de ésta, es decir, la información de la imagen conocida se utiliza para rellenar las áreas desconocidas.

Para producir una reconstrucción perceptiblemente plausible, las técnicas de Inpainting deben intentar continuar los isótopos (líneas de igual valor gris) lo más suavemente posible dentro de la región reconstruida.

Algunos de los métodos de Inpainting más conocidos, pero que no llegan a los resultados de los expuestos en este capítulo, serían los basados en el suavizado laplaciano de la imagen, como el propuesto por Bertalmio, el modelo de variación total, propuesto por Chan y Shen, o la difusión impulsada por la curvatura propuesta también por Chan y Shen, el cual mejora su anterior método de difusión a lo largo de las direcciones de los isótopos y que permite por lo tanto completar la información de regiones más amplias. Todos estos métodos lo que hacen, en esencia, es resolver una ecuación diferencial parcial o PDE, que describe la propagación del color dentro de la región carente de datos, la cual está sujeta a varios heurísticos que intentan preservar las direcciones de los isótopos; sin embargo, la preservación de los isótopos nunca se logra perfectamente en la práctica. Un ejemplo de este tipo de técnicas puede apreciarse en la imagen posterior.



Figura 5.5. Ejemplo de corrección mediante PDE
Fuente: elaboración propia, a partir de los datos del IGN

Los últimos enfoques de Inpainting que no utilizan Deep Learning emplean estadísticas de la imagen restante para llenar el hueco. Un ejemplo de ello sería PatchMatch, uno de los métodos de vanguardia, que busca iterativamente los parches más adecuados para colocar en los huecos existentes. Aunque este enfoque generalmente produce resultados suaves, es limitado por las estadísticas de imagen disponibles y no tiene ningún concepto de semántica visual, a diferencia de lo que ocurre con los modelos aquí desarrollados, por lo que los resultados obtenibles con PatchMatch no llegan a acercarse a la calidad de los obtenidos con métodos de Deep Learning, como el método de convoluciones parciales o los desarrollados durante esta tesis.

A pesar de que hay distintas formas de resolver el problema, hasta ahora (ya que en esta tesis se exploran otras alternativas mediante el uso de redes generativas antagónicas) todas dependen de la creación de una máscara, que localiza la posición de los píxeles problemáticos en la imagen original. Gracias a las inteligencias artificiales creadas, se puede prescindir de la fase de creación de máscaras, ya que es la propia Inteligencia Artificial la que detecta los errores en la imagen y genera una versión plausible y correcta de la misma.

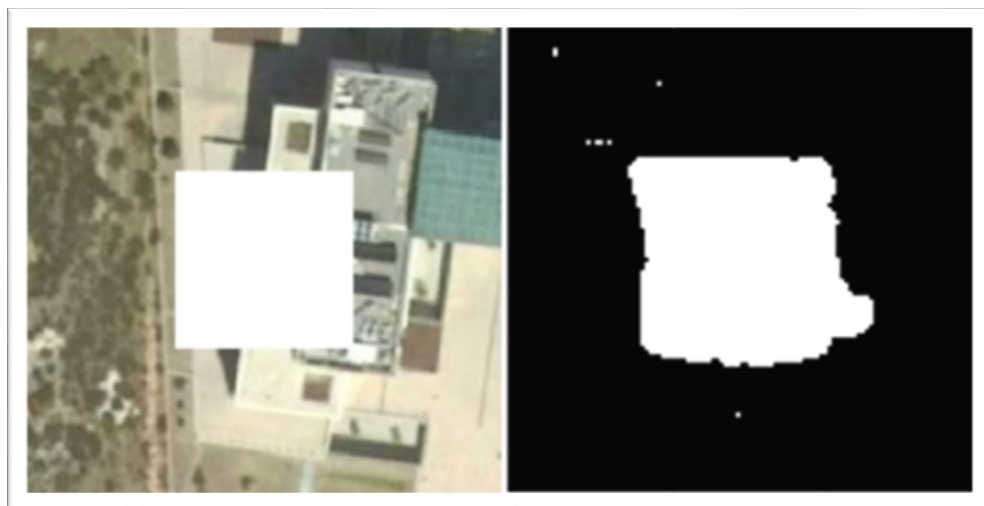


Figura 5.6. Generación manual de máscara
Fuente: elaboración propia, a partir de los datos del IGN

Para la creación de la máscara existen distintas técnicas. Tradicionalmente las máscaras eran creadas de forma manual, tal y como se muestra en las

siguientes imágenes del funcionamiento de uno de los scripts desarrollados para probar la eficacia de las técnicas ya existentes en las labores de Inpainting, el cual genera una simple interfaz gráfica a través de la cual el usuario puede generar la máscara.

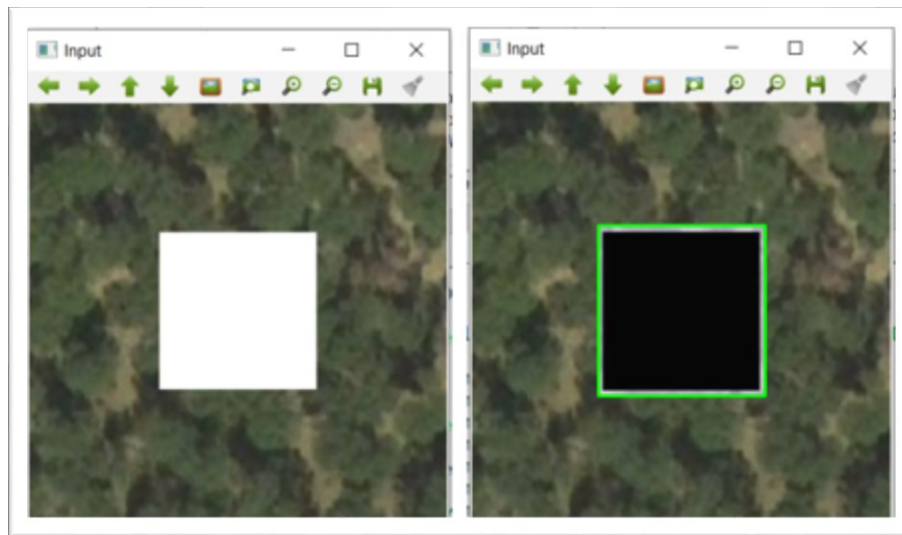


Figura 5.7. Generación manual de máscara
Fuente: elaboración propia, a partir de los datos del IGN

Pero en los últimos años, debido a los avances en visión por computadora, se han podido elaborar algoritmos que permiten la creación automática de máscaras, identificando los píxeles que se encuentran fuera de contexto. Para esta detección automática de anomalías y la posterior creación de una máscara, para su tratamiento y corrección, ha sido muy importante el desarrollo de librerías como OpenCV, la cual también ha sido de gran utilidad en el desarrollo de los distintos softwares y scripts. En el caso de la generación automática de máscaras aplicado a los scripts creados para mostrar el funcionamiento de las técnicas ya existentes, se han utilizado las funciones de OpenCV de detección de color, ajustando la tolerancia de ésta en función del caso concreto, tal y como se ve en la siguiente imagen.



Figura 5.8. Generación automatizada de máscara, a la izquierda la imagen original y a la derecha la máscara generada

Fuente: elaboración propia, a partir de los datos del IGN

Una vez localizado el problema, se aplican distintos algoritmos para la resolución de este; en los siguientes subepígrafes se comprueban los resultados ofrecidos por algunos de los más eficaces existentes en la actualidad.

5.2.1. Ecuación biarmónica

Este algoritmo realiza la reconstrucción de los píxeles faltantes o dañados de forma totalmente automática, aprovechando la información presente en las regiones no dañadas.

Como ya se ha mencionado anteriormente, en primer lugar, se genera una máscara de los píxeles dañados utilizando OpenCV, tras lo cual se aplica sobre los píxeles “enmascarados” el algoritmo basado en la ecuación biarmónica, la cual es ampliamente descrita en los trabajos de Hoang y Damelin, "On surface completion and image inpainting by biharmonic functions: numerical aspects".

Debe mencionarse, también, a este respecto el trabajo de Chui y Mhaskar, “MRA Contextual-Recovery Extension of Smooth Functions on Manifolds”. La descripción detallada de los principios matemáticos y topológicos en los que se basa el algoritmo utilizado en este subepígrafe se escapa al propósito de este documento, pero puede consultarse en la bibliografía en los dos trabajos anteriormente citados y en los artículos de Wikipedia sobre el algoritmo de “Inpainting” y la ecuación biarmónica.

A continuación, se muestran algunos ejemplos de la aplicación del algoritmo de “Inpainting”, basado en la ecuación biarmónica, a ortoimágenes aéreas extraídas como teselas de 256x256 píxeles de la hoja 0543 de la capa PNOA de Máxima Actualidad.

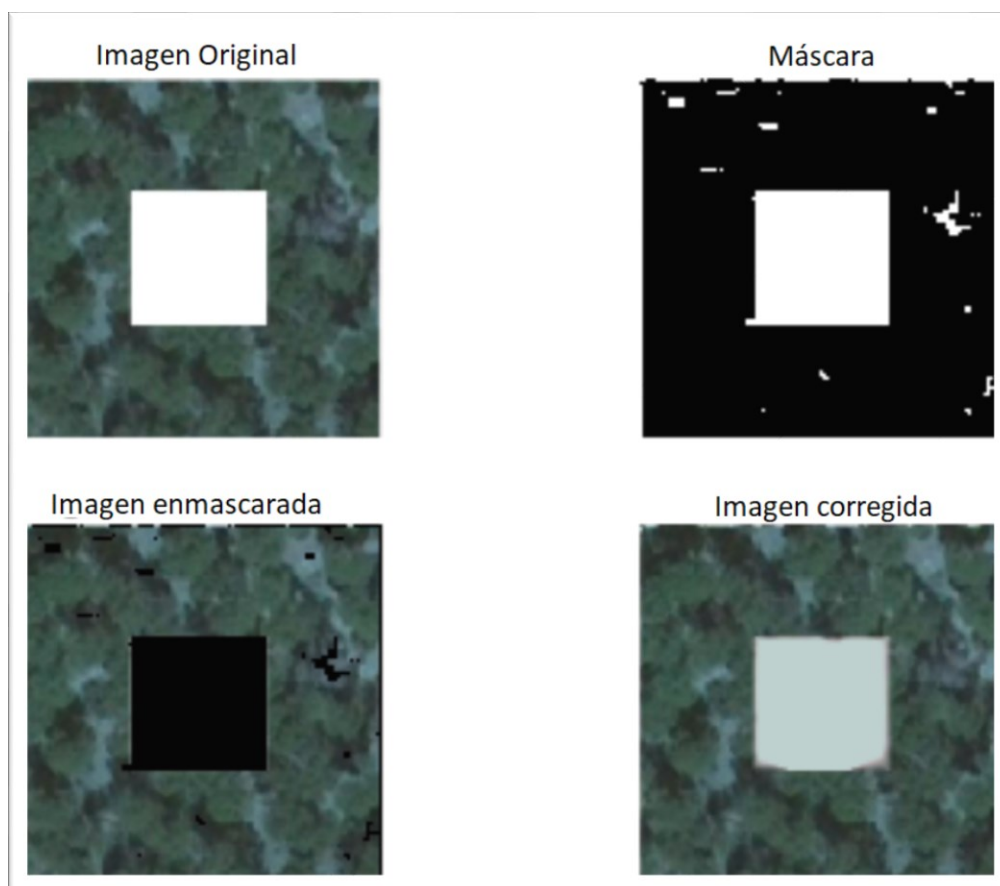


Figura 5.9. Corrección de huecos mediante la aplicación del método de ecuación biarmónica al área definida por la máscara generada de forma automática
Fuente: elaboración propia, a partir de los datos del IGN

Como puede apreciarse en los resultados obtenidos, la ecuación biarmónica no soluciona problemas en el caso de que los errores o artefactos ocupen una parte importante de la imagen a tratar. Debido a ello, debe recurrirse a otros sistemas como los basados en parches o directamente al Deep Learning, los cuales, para desarrollar todo su potencial, deben basarse en modelos entrenados con datos procedentes de ortoimágenes. Al no existir hasta la fecha este tipo de modelos, ha sido necesario crearlos expresamente para esta investigación. Seguidamente, puede comprobarse la diferencia entre los resultados obtenidos mediante la aplicación de la ecuación biarmónica, comparados con los obtenidos con las redes que se han entrenado.



Figura 5.10. Comparación de resultados resolviendo el mismo error de píxeles faltantes.

Fuente: elaboración propia, a partir de los datos del IGN

A pesar de que este método es ineficaz para realizar las labores de Inpainting en zonas amplias, puede resultar útil al aplicarse sobre imágenes en las que existe ruido producido por la ausencia de datos en determinados píxeles, simplemente adaptando la generación automatizada de la máscara a los valores carentes de datos, ya sean píxeles blancos o negros. La implementación de un script de Python que realice la generación automatizada de la máscara resulta sencilla utilizando la librería OpenCV, la cual posee funciones de detección de

color. A continuación, se muestra un ejemplo de la eliminación de los píxeles blancos del ruido de una imagen.

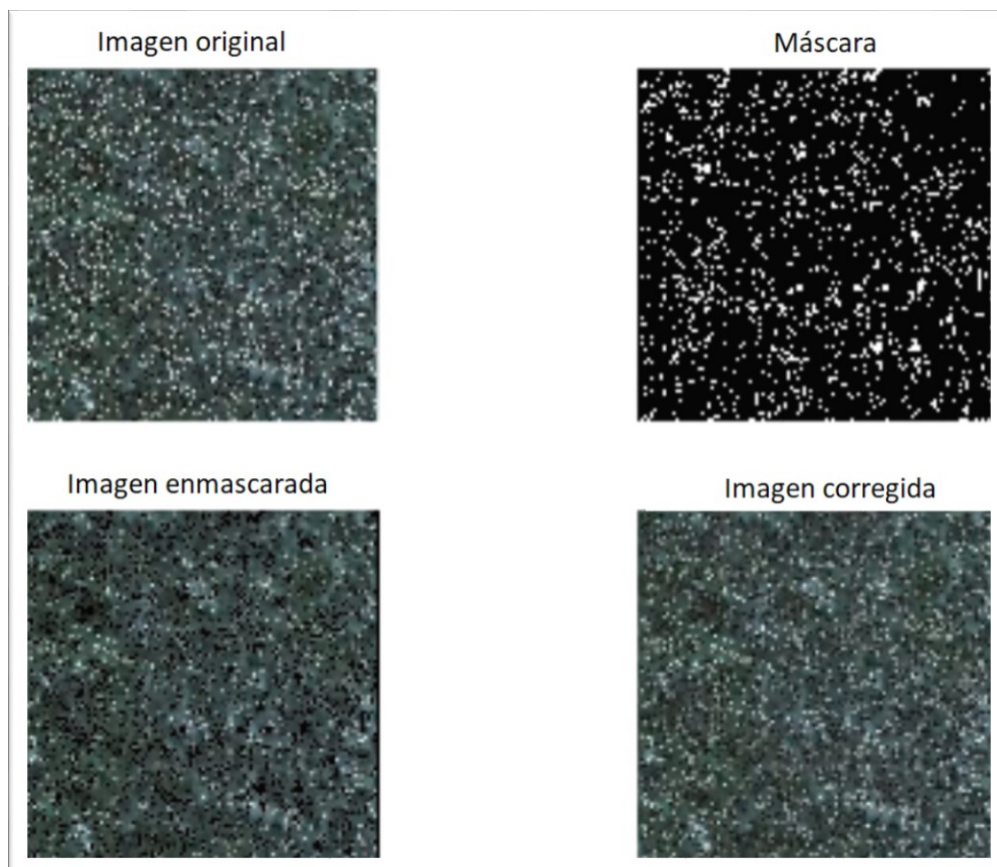


Figura 5.11. Ejemplo de resultado de eliminación de píxeles blancos mediante la aplicación de la ecuación biarmónica al área definida por la máscara generada de forma automática
Fuente: elaboración propia, a partir de los datos del IGN

Como puede apreciarse, a pesar de que algunos de los píxeles en blanco han sido corregidos, si el nivel de ruido es muy alto, como en esta imagen, el algoritmo es incapaz de asignar valores correctos a los nuevos píxeles generados. Debido a esta limitación de los métodos actuales para resolver problemas de ruido intenso, se han entrenado modelos capaces de eliminar el ruido intenso de forma automática. A continuación, se muestra la comparación de resultados al aplicar la ecuación biarmónica y el modelo “DeepMapsRuido”.



Figura 5.12. Comparación de resultados resolviendo el mismo error de ruido
Fuente: elaboración propia, a partir de los datos del IGN

Es interesante comprobar cómo este algoritmo puede aplicarse no sólo a ortoimágenes aéreas, sino también a mapas de elevaciones como el mostrado a continuación, extraído del modelo digital de elevaciones ASTER.

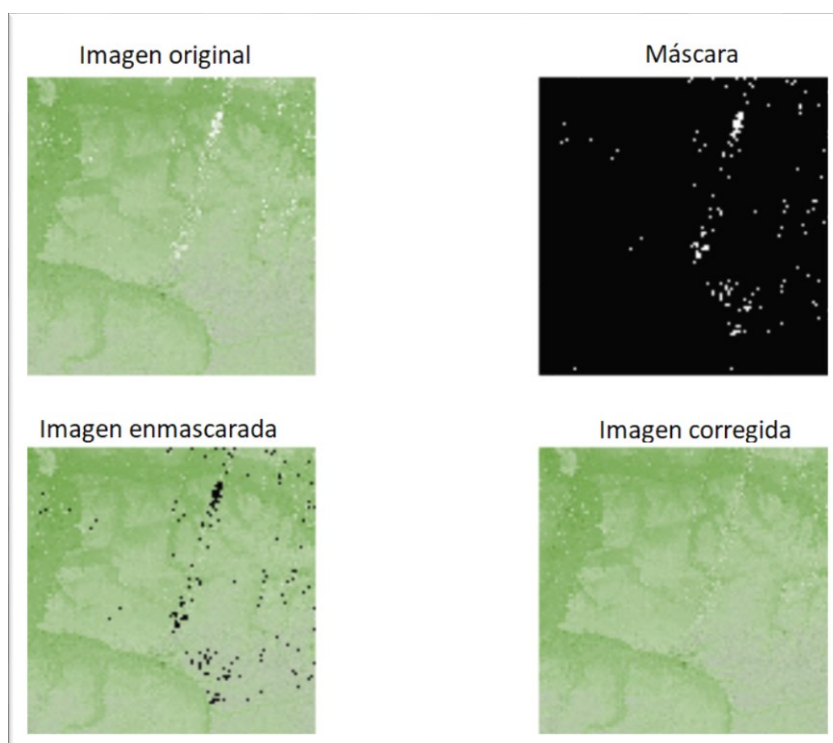


Figura 5.13. Corrección de píxeles blancos mediante la aplicación del método de ecuación biarmónica al área definida por la máscara generada de forma automática
Fuente: elaboración propia, a partir de los datos de Earthdata Search de la NASA.

Sin embargo, el hecho de corregir los errores en términos de imagen no soluciona del todo los mismos en su representación tridimensional, únicamente los atenúa; debido a ello, en esta investigación se han desarrollado métodos de corrección de los modelos digitales de elevaciones mediante reglas aplicadas a la comparación de los valores de altura, corrigiendo los datos tridimensionales de la nube de puntos a partir de la cual se crean. Todo este proceso, tanto de corrección como de generación del modelo digital de elevaciones, puede realizarse utilizando el software de elaboración propia “MEL” (Capítulos 8 y 9), el cual, además de proporcionar una resolución de los modelos mucho mayor a cualquier otra disponible actualmente (25 centímetros por píxel frente a los 2 metros por píxel del IGN), permite solucionar el problema de los píxeles con información errónea y los outliers. Para ello, en primer lugar MEL los identifica detectando alturas anómalas con respecto a los píxeles que hay alrededor y alturas anómalas con respecto al nivel del suelo; una vez detectados corrige la información de dichos píxeles (o puntos, en caso de operar directamente sobre la nube de puntos tridimensional), mediante una combinación entre interpolación bicúbica y la generación de una red irregular de triángulos o TIN para generar un modelo tridimensional a partir del cual se completa la información ausente según la geometría tridimensional generada. Este tipo de operaciones, que aprovechan las características tridimensionales de los datos LiDAR, no son aplicables a los datos bidimensionales como las ortoimágenes aéreas. Es por ello por lo que ha sido conveniente dividir esta tesis entre el software desarrollado para los problemas con datos procedentes de ortoimágenes aéreas, para las cuales se han entrenado inteligencias artificiales, y el software desarrollado para resolver las tareas relacionadas con datos tridimensionales, donde se han aplicado técnicas que aúnan interpolaciones ya utilizadas en el procesamiento de imágenes con otras más propias del modelado tridimensional. En ambos casos, para el desarrollo de inteligencias artificiales o para el software desarrollado para el tratamiento y manejo de datos tridimensionales, ha sido necesario aplicar técnicas propias del Big Data, debido al volumen de datos que era necesario procesar, el cual es inabarcable con metodologías y tecnologías tradicionales.

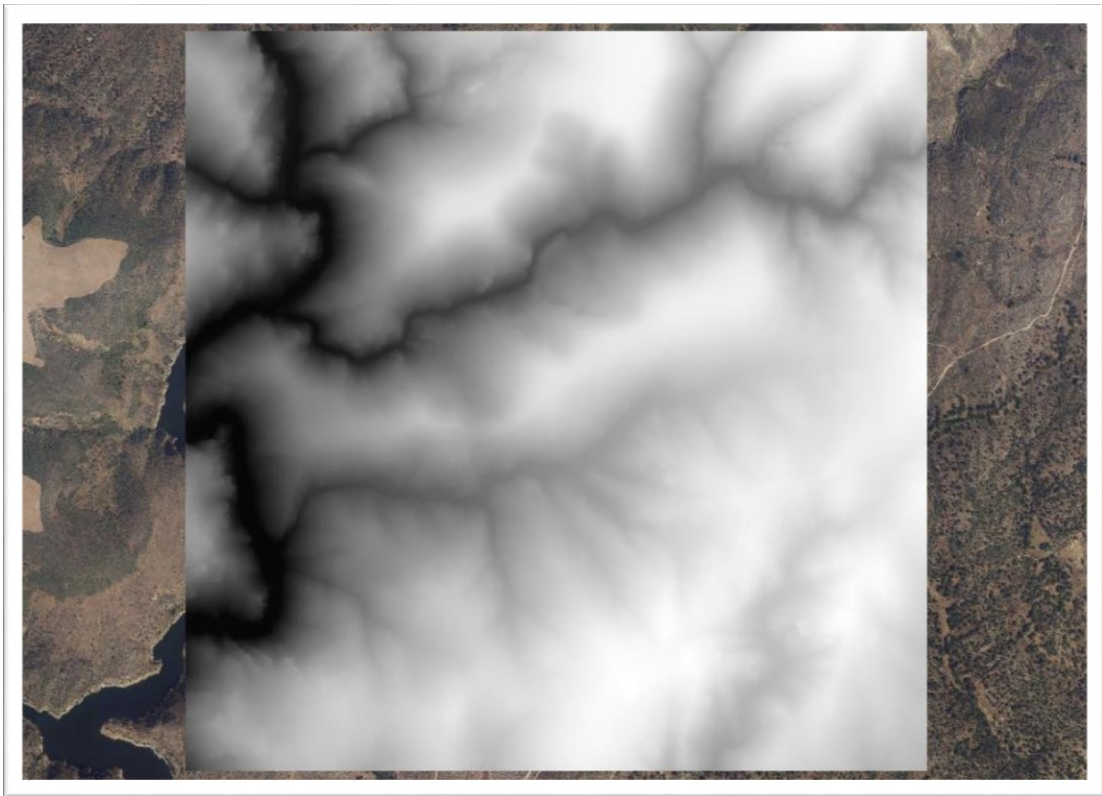


Figura 5.14. Modelo digital de elevaciones con malla de 25 cm de resolución, obtenido con el software MEL, a partir de los datos LiDAR del archivo de nube de puntos: PNOA_2009_Lote6_CLM_498-4546_ORT-CLA-COL.laz del IGN sobre ortoimagen de Máxima Actualidad del PNOA. EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)
Fuente: elaboración propia a partir de los datos del IGN

5.2.2. Método de marcha rápida

Los resultados que se muestran a continuación están basados en la técnica descrita por Alexandru Telea en su publicación de 2004 “An image inpainting technique based on the fast marching method”. Este método, al igual que la ecuación biarmónica comentada en el subepígrafe anterior, no se basa en redes neuronales, sino en la aplicación de algoritmos ya existentes para solucionar el problema de completar la información ausente. Debido a esto, tanto la técnica anterior como ésta, son especialmente rápidas y por lo tanto deberían ser consideradas como una opción rápida a la hora de reparar pequeños daños en las ortoimágenes aéreas; sin embargo, los resultados obtenidos no son

satisfactorios para toda la casuística posible, ya que como puede verse en los ejemplos mostrados en este subepígrafe, el método resulta ineficaz cuando las zonas a reparar son amplias.

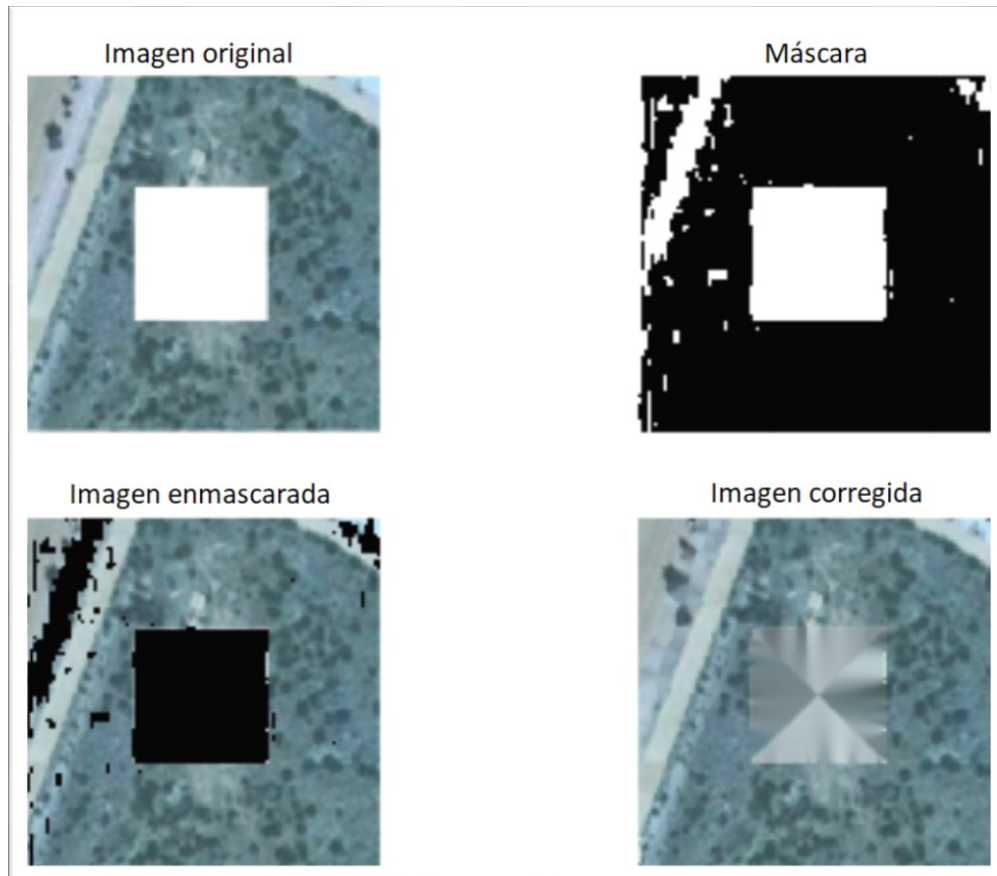


Figura 5.15. Proceso de corrección de huecos mediante la aplicación del método de marcha rápida al área definida por la máscara generada de forma automática
Fuente: elaboración propia, a partir de los datos del IGN

Este algoritmo es muy simple de implementar, rápido, y produce resultados casi idénticos a los métodos conocidos más complejos y generalmente más lentos. Debido a la eficiencia de este método, su aplicación estaría indicada para reparaciones de extensiones pequeñas de píxeles; sin embargo, cuando el área a generar supera cierto tamaño no resulta útil, tal y como puede verse en el ejemplo ilustrado en la figura 5.15. A continuación, se muestra una comparación de los resultados obtenidos con el método de marcha rápida frente a la imagen generada con la inteligencia artificial DeepMapsInpainting.



Figura 5.16. Comparativa de resultados resolviendo el mismo error completando los datos faltantes mediante el método de marcha rápida y mediante la aplicación del modelo DeepMapsInpainting

Fuente: elaboración propia, a partir de los datos del IGN

Como puede comprobarse, los resultados obtenidos con este método están muy lejos de alcanzar el nivel de perfección proporcionado por los modelos entrenados, como es el caso de DeepMapsInpainting.

5.2.3. Convoluciones parciales con una actualización automática de la máscara

Por regla general, los métodos basados en Deep Learning ofrecen mejores resultados que los tradicionales en la tarea de Inpainting, siempre que los modelos hayan sido convenientemente entrenados. Dentro de estos métodos de Inpainting, aplicando Inteligencia Artificial, cabe destacar el método de convoluciones parciales.

Este método supone uno de los más novedosos en la actualidad y ofrece mejores resultados que los anteriores métodos basados en Deep Learning. Sin embargo, al no haber sido entrenadas estas redes con datos procedentes de ortoimágenes aéreas, los resultados al aplicarse en este tipo de dominios no alcanzan el nivel de perfección de los desarrollados en esta investigación en lo que a las labores cartográficas se refiere. Los métodos de corrección de artefactos anteriores utilizan una red convolucional estándar sobre la imagen corrupta, usando respuestas de filtro convolucional condicionadas, tanto a los píxeles válidos como a los valores de sustitución en los agujeros enmascarados (típicamente el valor medio). Éste a menudo conduce a errores como la discrepancia de color y la falta de nitidez. Estos errores pueden solventarse, en cierta medida, mediante el postprocesamiento, pero es costoso y puede fallar; es por ello por lo que los modelos aquí desarrollados evitan también esta costosa fase de postprocesamiento. Este método propone el uso de convoluciones parciales, donde la convolución se enmascara y se renormaliza para ser condicionada sólo a los píxeles válidos; tras esto se actualiza y se repite el proceso. En el paso de actualización automática de la máscara, que elimina cualquier enmascaramiento en el que la convolución parcial haya podido operar con un valor sin enmascarar, dadas las suficientes capas de actualizaciones sucesivas, incluso los mayores agujeros enmascarados se encogen, dejando sólo respuestas válidas en el mapa de características.



Figura 5.17. Ejemplos de la aplicación del método de convoluciones parciales con actualización automática de la máscara

Fuente: elaboración propia a partir de los datos del IGN

Comparativa de resultados:

A continuación, se muestran los resultados obtenidos con este método de Inpainting (convoluciones parciales con actualización automática de la máscara) que, dentro de los métodos ya existentes, es el que mejor desempeño ha realizado hasta ahora, en comparación con los resultados entregados por la inteligencia artificial DeepMapsInpainting.



Figura 5.18. Imágenes de entrada utilizadas en las comparativas siguientes
Fuente: elaboración propia, a partir de los datos del IGN

Resultados:

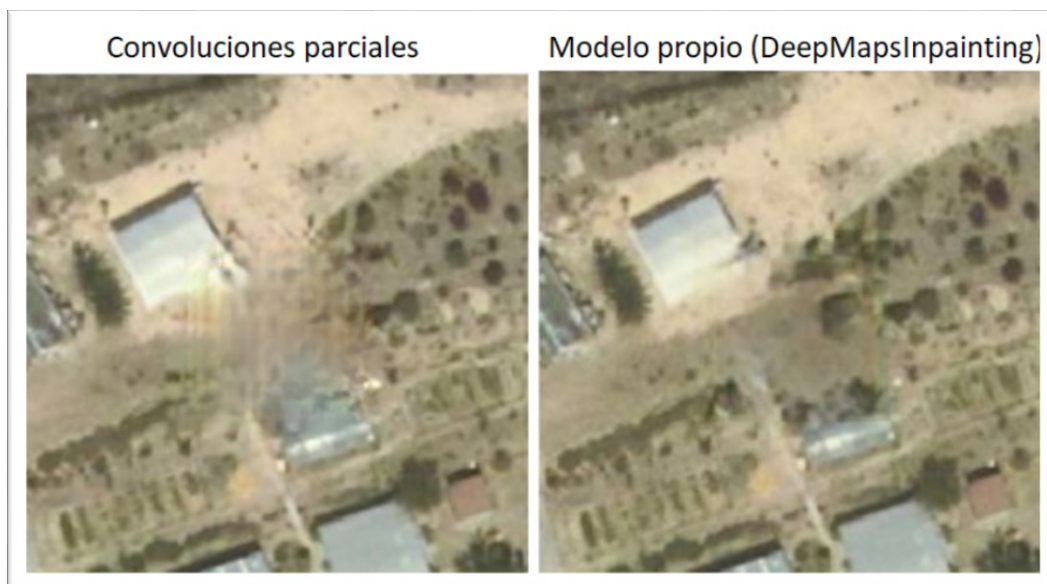


Figura 5.19. Comparación de resultados al aplicar el método de convoluciones parciales con actualización automática de máscara (izquierda) y el modelo DeepMapsInpainting (derecha)
Fuente: elaboración propia, a partir de los datos del IGN



Figura 5.20. Comparación de resultados al aplicar el método de convoluciones parciales con actualización automática de máscara (izquierda) y el modelo DeepMapsInpainting (derecha)
Fuente: elaboración propia, a partir de los datos del IGN

Como puede comprobarse, el modelo entrenado con datos de ortoimágenes aéreas ofrece resultados muy superiores a todo lo desarrollado con anterioridad, en cuanto a datos cartográficos se refiere.

5.3. Modelos propios

Como ya se mencionó, en cualquier imagen digital la medición de los tres valores de color observados en cada píxel está sujeta a algunas perturbaciones. Estas perturbaciones se deben a la naturaleza aleatoria del conteo de fotones en cada sensor. El ruido puede ser amplificado por las correcciones digitales de la cámara o por cualquier software de procesamiento de imágenes; para resolver esta cuestión se han entrenado los modelos DeepMapsRuido y DeepMapsRuidoDesenfoco que se exponen en los próximos dos subepígrafes.



Figura 5.21. Ejemplo de aplicación del modelo propio DeepMapsRuidoDesenfoco
Fuente: elaboración propia, a partir de los datos del IGN

Debe mencionarse en este capítulo que los errores y artefactos que afectan a la coloración de las imágenes, pueden resolverse aplicando los modelos aquí desarrollados para la coloración automatizada de imágenes, los cuales fueron expuestos en el capítulo 3; sin embargo, en base a los conocimientos obtenidos, gracias a las múltiples pruebas realizadas durante el desarrollo de esta investigación, se han creado algunas inteligencias artificiales

capaces de corregir numerosos problemas presentes en una misma ortoimagen. Si bien es cierto que una inteligencia artificial entrenada en una única tarea resulta más eficaz en el desempeño de la misma que una que ha sido entrenada para realizar diversas tareas, en los casos prácticos reales los errores que se presentan no suelen ser tan graves como aquellos para los que se han entrenado las inteligencias artificiales más específicas (sirviendo estas, no obstante para mostrar las posibilidades y la capacidad de las técnicas propuestas en este trabajo). Debido a ello se han creado los modelos DeepMapsRestaura, los cuales son capaces de corregir todos los problemas habituales de las ortoimágenes aéreas, como son la luminosidad, el contraste, el color, el ruido, el desenfoque, la existencia de errores producidos por fallos en la digitalización, presencia de partículas como polvo o pelos, roturas del soporte físico, etc. En los últimos subepígrafes de este capítulo se incluye una mención más detallada de los modelos DeepMapsRestarua, y en el capítulo 7 puede comprobarse su eficacia al ser aplicados a casos de uso reales.



Figura 5.22. Reparación de ortoimagen procedente del Vuelo Interministerial (1973-1986) mediante la aplicación del modelo propio DeepMapsRestaura¹⁷
Fuente: elaboración propia, a partir de los datos del IGN

Por otra parte, debido a los buenos resultados obtenidos con las inteligencias artificiales de creación propia para la corrección de errores mediante la generación de datos plausibles que sustituyan los datos dañados, sus aplicaciones van más allá de la reparación de ortoimágenes, ya que gracias a su funcionamiento, pueden ser utilizadas también para eliminar elementos presentes en las imágenes sin que sea apreciable la manipulación de las mismas, rellenando el espacio con datos plausibles y realistas.



Figura 5.23. Eliminación de elementos utilizando la inteligencia artificial “DeepMapsInpainting”
Fuente: elaboración propia, a partir de los datos del IGN

En la imagen superior puede observarse como, mediante la aplicación del modelo DeepMapsInpainting desarrollado en esta investigación, puede eliminarse un edificio de manera automática sin que sea apreciable la modificación. Hasta ahora la única manera de realizar esta tarea era mediante la edición manual de las imágenes, realizada por especialistas en edición gráfica y los resultados, normalmente, no resultaban tan satisfactorios.

A continuación, se presentan varios de los modelos de Inteligencia Artificial de elaboración propia y los resultados que se han obtenido al ser aplicados a datos con las mismas características con las que se probaron los métodos ya existentes en los epígrafes anteriores, para así poder comprobar las diferencias en los resultados.

En primer lugar, se analizan los modelos entrenados para la eliminación de ruido y, después, el modelo entrenado para labores de Inpainting, con el cual pueden corregirse multitud de errores y artefactos de diversos tipos.

5.3.1. DeepMapsRuido

Este modelo ha sido entrenado para la eliminación del ruido en las ortoimágenes aéreas, mediante un total de 5.712 imágenes seleccionadas de las hojas de la Comunidad de Madrid del PNOA de Máxima Actualidad, las cuales han sido tratadas mediante un script de Python para añadir ruido en forma de píxeles blancos y negros, insertados de forma aleatoria en la imagen original y generar el par de imágenes necesario para la ingesta en el modelo.

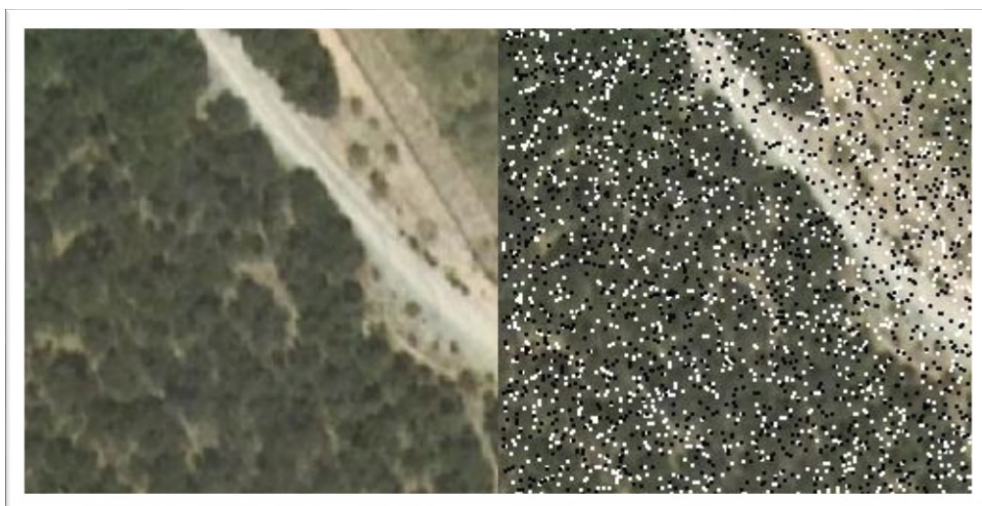


Figura 5.24. Ejemplo de uno de los pares de imágenes utilizados para el entrenamiento del modelo DeepMapsRuido

Fuente: elaboración propia, a partir de los datos del IGN

El entrenamiento se ha prolongado durante 10.000 iteraciones y ha producido los siguientes resultados.



Figura 5.25. Resultados obtenidos con el modelo DeepMapsRuido
Fuente: elaboración propia, a partir de los datos del IGN

5.3.2. DeepMapsRuidoyDesenfoque

Este modelo ha sido entrenado para realizar dos operaciones simultáneamente, la eliminación del ruido y el enfoque de la imagen; para ello se ha partido de teselas de 256x256 píxeles, extraídas de las hojas del PNOA de Máxima Actualidad correspondientes a la Comunidad de Madrid; sobre estas imágenes se ha aplicado un desenfoque gaussiano con un radio de 4 píxeles y se ha incluido ruido, para de esta manera crear los pares de imágenes necesarios para el entrenamiento.

En el caso de esta Inteligencia Artificial, el entrenamiento se ha realizado con 1.594 imágenes durante 4.500 iteraciones.



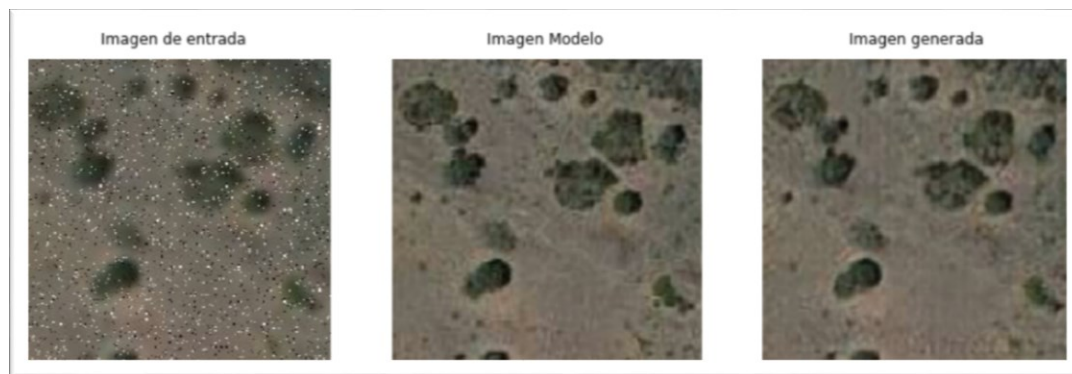


Figura 5.26. Resultados obtenidos con el modelo DeepMapsRuido y Desenfoque
Fuente: elaboración propia, a partir de los datos del IGN

Esta Inteligencia Artificial se ha entrenado para ser capaz de corregir el desenfoque y el ruido presente en las ortoimágenes aéreas en color, tal y como se muestra en el siguiente ejemplo.

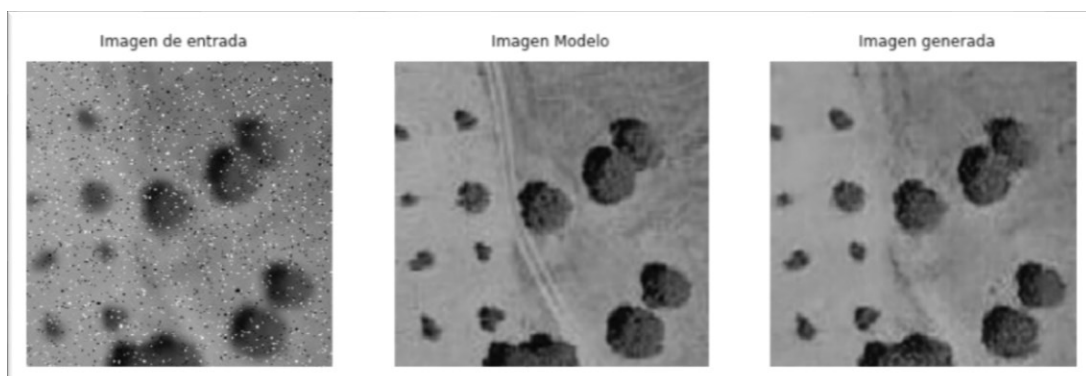


Figura 5.27. Corrección de ortoimagen realizada con el modelo DeepMapsRuido y Desenfoque
Fuente: elaboración propia, a partir de los datos del IGN

Como queda constatado mediante los resultados expuestos anteriormente, las capacidades de corrección de imágenes desarrolladas por las inteligencias artificiales entrenadas en este estudio superan ampliamente a las de las técnicas existentes hasta ahora.

5.3.3. DeepMapsRuidoyDesenfoqueBN

Esta inteligencia artificial es similar a la expuesta en el subepígrafe anterior; las imágenes con las que ha sido entrenada, además de haber sido preprocesadas previamente con un script de Python que incluye un desenfoque gaussiano con un radio de 4 píxeles y ruido. En este caso, se ha modificado el script utilizado para la generación de los datos de entrenamiento del modelo DeepMapsRuidoyDesenfoqueBN, para que extrajese el canal L del espacio de color $L^*a^*b^*$, y de esta manera generar imágenes en escala de grises las cuales permiten entrenar el modelo para la eliminación del ruido y el enfoque de la imagen, sin preocuparse por el color. Esto hace al modelo especialmente adecuado para la reparación de defectos en ortoimágenes antiguas en blanco y negro. Las ortoimágenes originales utilizadas en el entrenamiento son teselas de 256x256 píxeles, extraídas de ortoimágenes aéreas del PNOA de Máxima Actualidad, a partir del wms del IGN (<http://www.ign.es/wms-inspire/pnoa-ma?request=GetCapabilities&service=WMS>). Asimismo, estas imágenes han sido tratadas con el script descrito anteriormente, para, de esta manera, crear los pares de imágenes necesarios para el entrenamiento. El modelo ha sido entrenado durante 3.200 iteraciones con un total de 2.328 pares de imágenes.



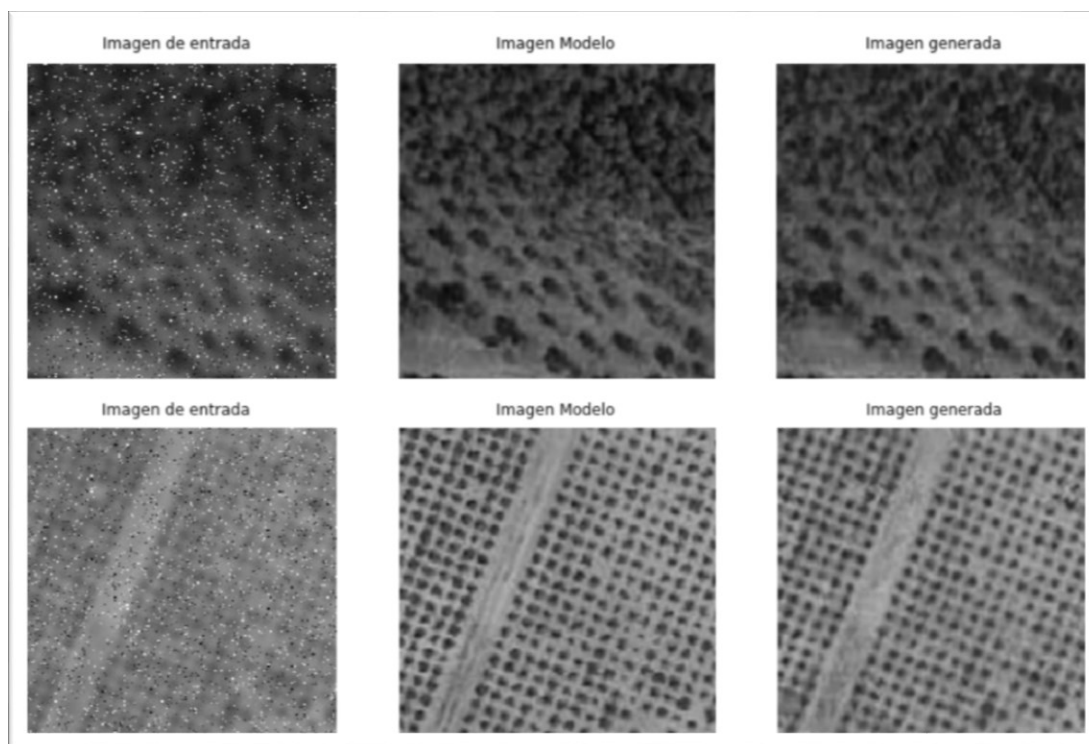


Figura 5.28. Resultados de la aplicación del modelo DeepMapsRuidoyDesenfoqueBN frente a la imagen original (Imagen Modelo)

Fuente: elaboración propia, a partir de los datos del IGN

5.3.4. DeepMapsInpainting

Esta Inteligencia Artificial ha sido creada con el propósito de realizar labores de Inpainting sobre ortoimágenes aéreas, es decir, poder generar los datos faltantes en una imagen o reemplazar los datos erróneos, pudiendo, por lo tanto, corregir cualquier tipo de artefacto o, incluso, eliminar elementos presentes en la imagen sin que se aprecie la intervención, ya que la información es reemplazada por información plausible. El modelo DeepMapsInpainting, a pesar de haber sido entrenado con imágenes a color y parches blancos de dimensiones regulares, en las pruebas realizadas ha demostrado ser capaz de añadir información verosímil a ortoimágenes que presentaban parches irregulares, respetando la paleta de colores de la ortoimagen a tratar, tal y como puede apreciarse en el siguiente ejemplo, en el que se ha reparado una ortoimagen aérea del vuelo OLISTAT.

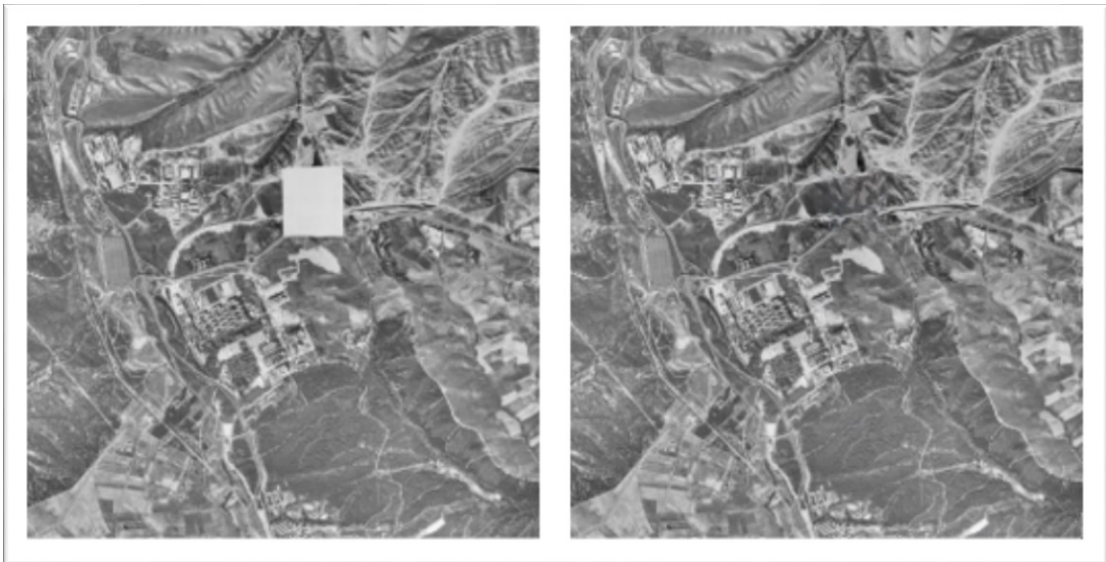


Figura 5.29. Ejemplo de corrección de datos inexistentes debido a la censura de una parte de la base militar El Goloso - Briac XII. (Escala: 1:25.000)
Fuente: elaboración propia, a partir de los datos del IGN

Para el entrenamiento de este modelo se han utilizado teselas de 256x256 píxeles, con un nivel de zoom de 19, o lo que es lo mismo una escala de 1:1.000, extraídas de las hojas correspondientes a la Comunidad de Madrid del PNOA de Máxima Actualidad. Los pares de imágenes necesarios para el entrenamiento han sido generados mediante un script de Python, que elimina la información del centro de las teselas, concretamente un rectángulo que abarca desde el píxel 80, 80 al píxel 175,175.



Figura 5.30. Ejemplo de uno de los pares de imágenes generados para el entrenamiento del modelo DeepMapsInpainting
Fuente: elaboración propia, a partir de los datos del IGN

En entrenamiento se ha realizado con 3.200 imágenes durante 9.000 iteraciones.

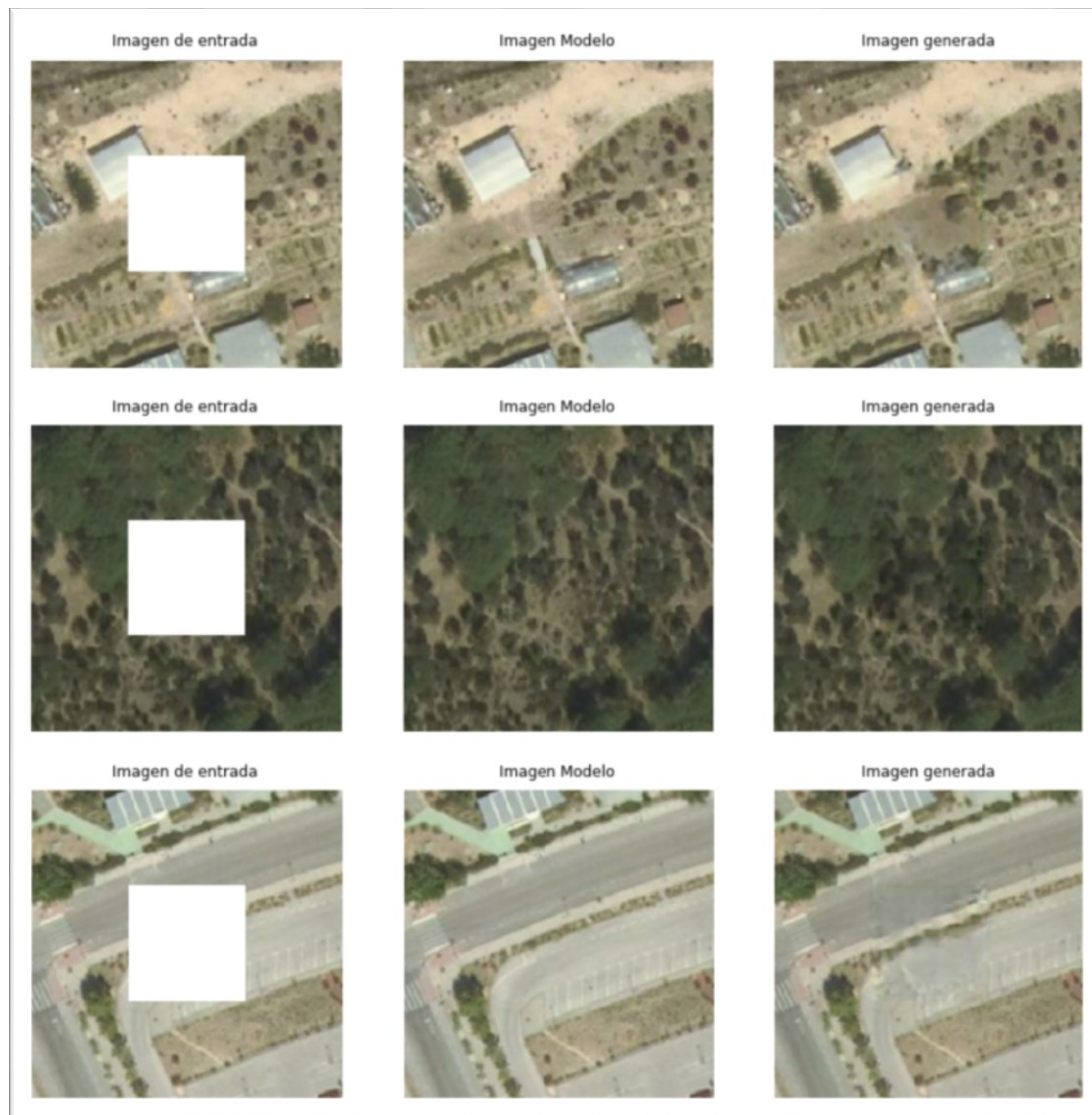


Figura 5.31. Resultados obtenidos con el modelo DeepMapsInpainting frente a la imagen original

Fuente: elaboración propia, a partir de los datos del IGN

Tal y como puede comprobarse en los resultados expuestos anteriormente y en los ejemplos mostrados más en detalle a continuación, la Inteligencia Artificial DeepMapsRestarua es capaz de completar los datos faltantes con datos verosímiles, tanto en entornos naturales como de origen antrópico, de forma mucho más eficaz que cualquiera de los algoritmos o inteligencias

artificiales existentes en la actualidad, al menos en lo que a ortofotografías aéreas se refiere.

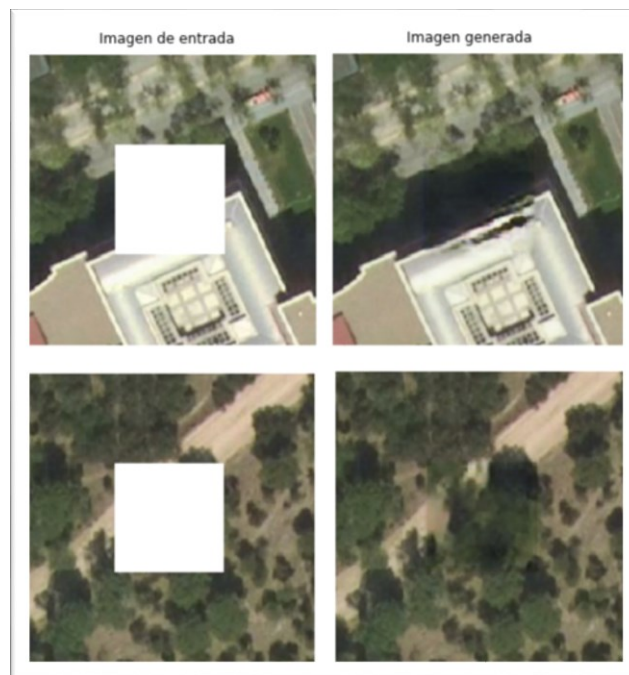


Figura 3.32. Ejemplo de resultados obtenidos con el modelo DeepMapsInpainting
Fuente: elaboración propia, a partir de los datos del IGN

5.3.5. DeepMapsNubes

Esta red ha sido entrenada con 1.532 imágenes seleccionadas de forma aleatoria de la totalidad del territorio español, mediante un script de Python, desarrollado para tal fin, a partir de ortoimágenes del Landsat 8 con 15 metros de resolución de píxel obtenidas del IGN. Para crear los datos de entrenamiento se han utilizado teselas de 256x256 píxeles generadas a partir de las ortoimágenes del Landsat con un nivel de zoom de 13 (escala aproximada de 1:70.000). En este caso, con el propósito de poder entrenar una inteligencia artificial para la eliminación de nubes se ha recurrido a la creación de un script de Python que edite las imágenes originales del Landsat para generar las imágenes de entrada del modelo, añadiendo a las ortoimágenes originales ruido aleatorio y elipses blancas y negras, de posición y tamaño aleatorio para simular

la existencia de nubes y sombras. En cuanto a las imágenes originales utilizadas, de la selección aleatoria original se eliminaron aquellas que contenían errores o presencia de nubes, para mejorar la efectividad del modelo aumentando la calidad de los datos de entrenamiento. El número de iteraciones durante las que ha sido entrenado este modelo asciende a 3.200.

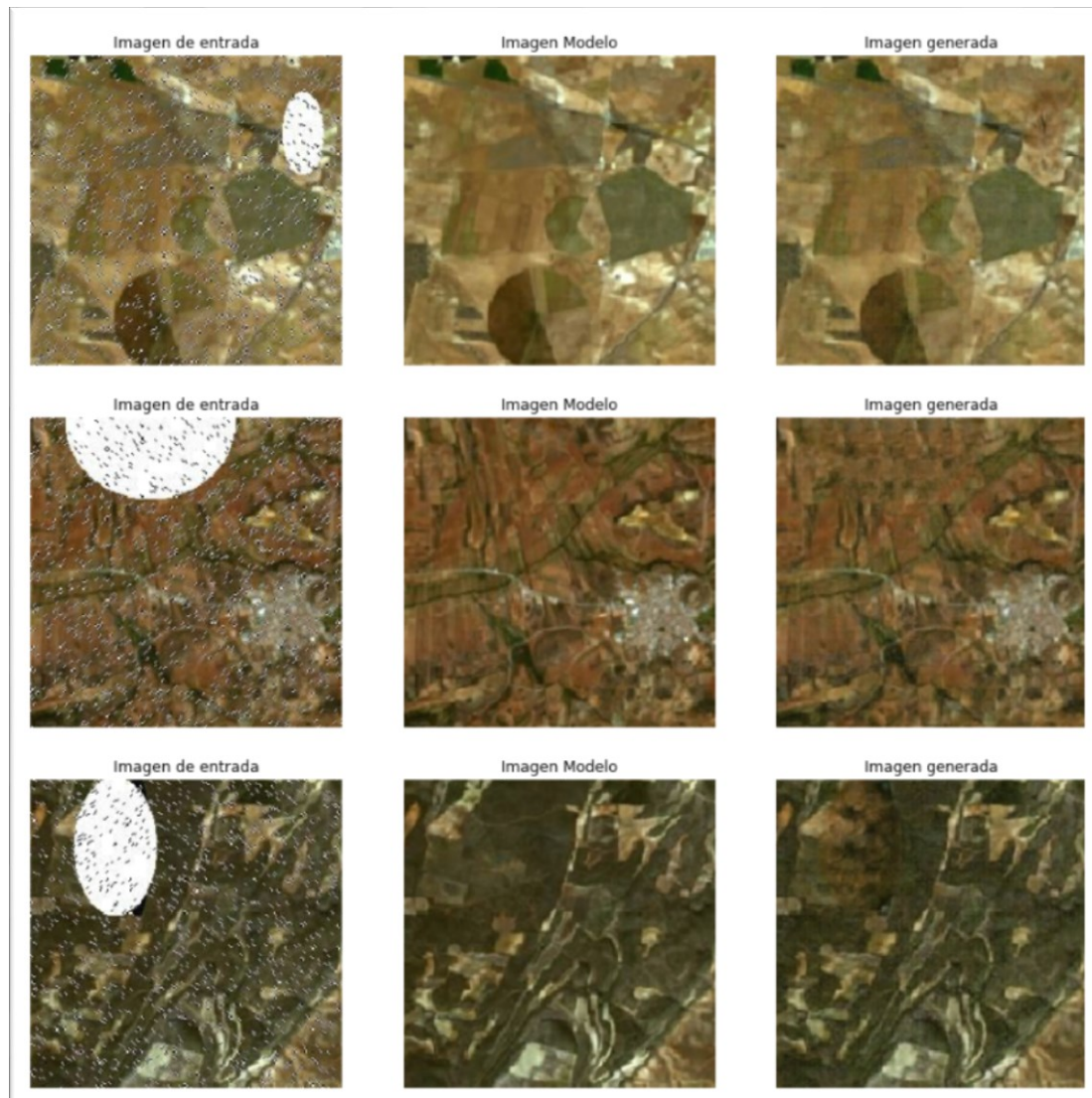


Figura 5.33. Resultados de la aplicación del modelo DeepMapsNubes frente a la imagen original (Imagen Modelo)

Fuente: elaboración propia, a partir de los datos del IGN

5.3.6. DeepMapsReparaHipsométrico

Esta Inteligencia Artificial ha sido entrenada, específicamente, para eliminar los parches blancos existentes en las ortoimágenes del mapa hipsométrico del SCUAM; el nivel de zoom para el que está optimizada es de 13, es decir, una escala aproximada de 1:70.000.

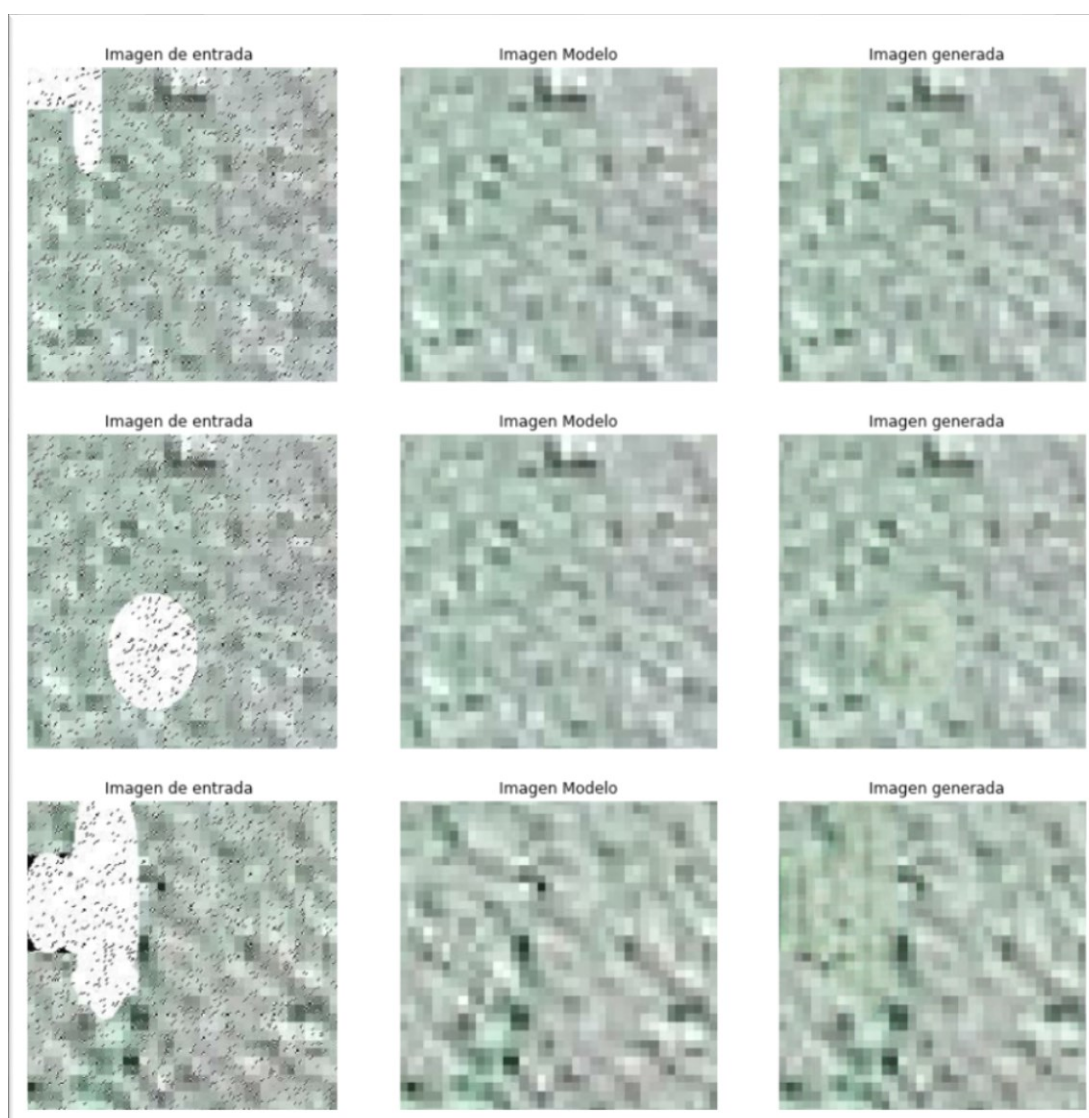


Figura 5.34. Resultados de la aplicación del modelo DeepMapsRestauraHipsométrico frente a la imagen original (Imagen Modelo)

Fuente: elaboración propia, a partir de los datos del IGN

5.3.7. DeepMapsRestaura15

Este modelo ha sido entrenado para realizar varias tareas de forma simultánea, incluyendo la coloración, la eliminación de ruido, la eliminación de desperfectos, la corrección del brillo, la corrección de contraste, la mejora de la resolución, la corrección del desenfoque y la mejora de la nitidez en general, detectando las deficiencias de las ortoimágenes originales y aplicando las distintas correcciones, en el grado necesario, en función de los datos a tratar de forma automática, para proporcionar una salida con características similares a las ortoimágenes del PNOA de Máxima Actualidad.

Para el entrenamiento de esta inteligencia artificial, se han utilizado 17.200 pares de imágenes y el entrenamiento ha durado 14.800 iteraciones. Las ortoimágenes originales han sido extraídas de ortofotografías aéreas del PNOA de Máxima Actualidad, procedentes del IGN, a partir de las cuales se han generado teselas de 256x256 píxeles con un nivel de zoom de 15, es decir una escala aproximada de 1:15:000. Para la generación de las imágenes de entrada y la construcción de los pares de imágenes utilizados en el entrenamiento, se ha recurrido a la creación de teselas derivadas de la original, en las que se ha modificado la luminosidad y el contraste, además de extraer el canal L del espacio de color $L^*a^*b^*$, como se ha hecho ya en el resto de los modelos entrenados para dar color a ortoimágenes en blanco y negro.

Este tipo de modificaciones son fundamentales en el trabajo cartográfico, y, debido a que hasta ahora no se habían aplicado las redes neuronales a las labores cartográficas expuestas en esta tesis, nadie se había planteado la necesidad de variar la luminosidad y el contraste en este procesado previo de las imágenes originales. La generación de distintas combinaciones de luminosidad y contraste no sólo permite aumentar los datos de entrenamiento, sino que aporta una gran versatilidad a las redes entrenadas de esta forma, lo cual es especialmente importante al trabajar con ortoimágenes antiguas, ya que estas presentan una gran variabilidad en estos factores. Además de las

variaciones de la luminosidad y el contraste, se han incluido en los datos de entrenamiento variaciones del enfoque con un desenfoque gaussiano de 0 a 2 píxeles, lo cual permite que el modelo mejore la nitidez de las imágenes originales. Este preprocesamiento concreto de las imágenes utilizadas para el entrenamiento de una red neuronal, en el que se generan decenas de variaciones de una misma imagen, es una innovación no sólo en el campo de las redes neuronales aplicadas a la cartografía, sino en todo lo que a redes convolucionales se refiere.



Figura 5.35. Resultados de la aplicación del modelo DeepMapsRestaura15 frente a la imagen original (Imagen Modelo)

Fuente: elaboración propia a partir de los datos del IGN

Tal y como se aprecia en las imágenes anteriores, el modelo es capaz de corregir las deficiencias producidas por el desenfoque y las variaciones de brillo y contraste de las ortoimágenes de entrada, generando ortoimágenes muy similares a la original (Imagen Modelo).

Además de la red DeepMapsRestaura15, siguiendo el mismo procedimiento se han creado otras dos redes optimizadas para distintas escalas, DeepMapsRestarua17 y DeepMapsRestarua19.

5.3.8. DeepMapsRestaura17

El entrenamiento de esa red ha sido similar al entrenamiento de la red DeepMapsRestaura15, con variaciones de las imágenes originales, en las que se ha extraído el canal L del espacio $L^*a^*b^*$, se ha cambiado el brillo, el contraste y añadido un desenfoque gaussiano, el cual, a diferencia del modelo anterior, tiene un radio de desenfoque que va desde 0 a 4 píxeles. Otra diferencia con respecto al modelo anterior es que el set de ortoimágenes utilizadas para el entrenamiento fue diferente y las teselas extraídas de los datos originales del PNOA de Máxima Actualidad tienen un nivel de zoom 17 que equivale a una escala aproximada de 1:4.000, en lugar de 1:15.000. Este modelo ha sido entrenado con 12.600 pares de imágenes durante un total de 5.300 iteraciones. A continuación, se expone una muestra del funcionamiento de este modelo.



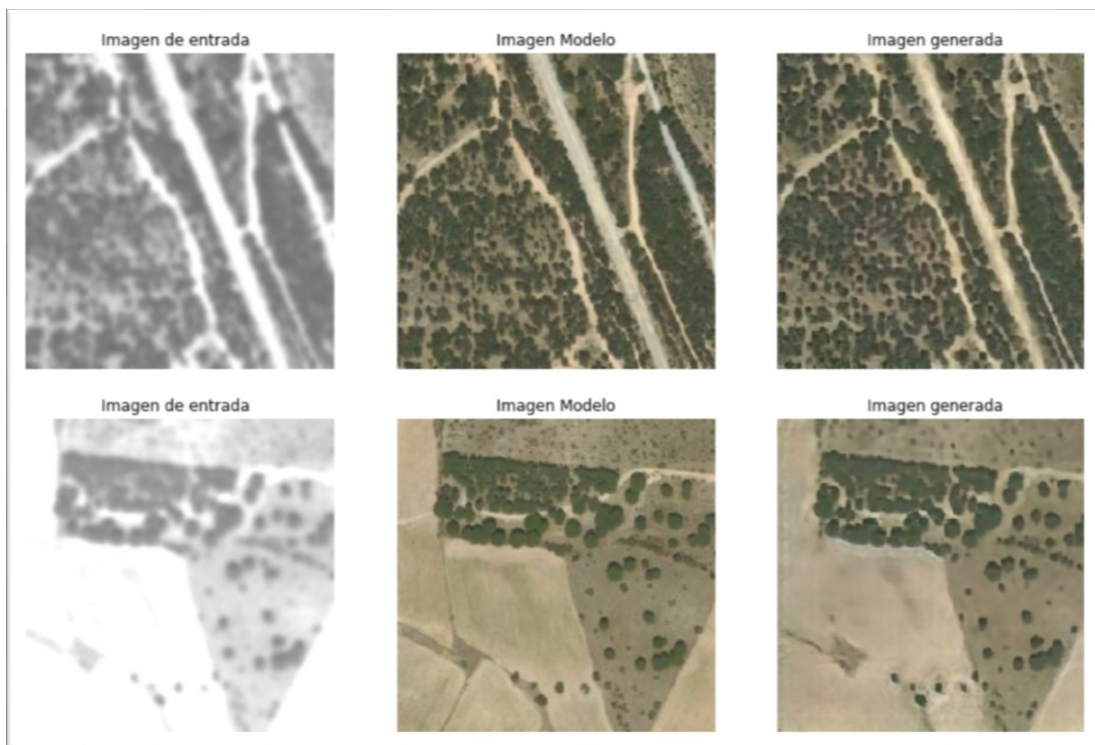


Figura 5.36. Resultados de la aplicación del modelo DeepMapsRestaura17 frente a la imagen original (Imagen Modelo)

Fuente: elaboración propia, a partir de los datos del IGN

En los ejemplos mostrados sobre este párrafo, se puede comprobar como esta Inteligencia Artificial es capaz de reconstruir fielmente ortoimágenes muy deterioradas, en las que incluso se ha perdido una parte considerable de la información, lo cual resulta muy útil a la hora de reparar deterioros en ortoimágenes antiguas.

5.3.9. DeepMapsRestaura19

Esta Inteligencia Artificial, al igual que las dos anteriores, ha sido entrenada con ortoimágenes derivadas de las originales a partir de variaciones del brillo y el contraste, el desenfoque (con un desenfoque gaussiano de 1 a 6 píxeles) y la extracción del canal L del espacio de color $L^*a^*b^*$. Para este entrenamiento se han utilizado 22.400 pares de imágenes durante 8.200 iteraciones.

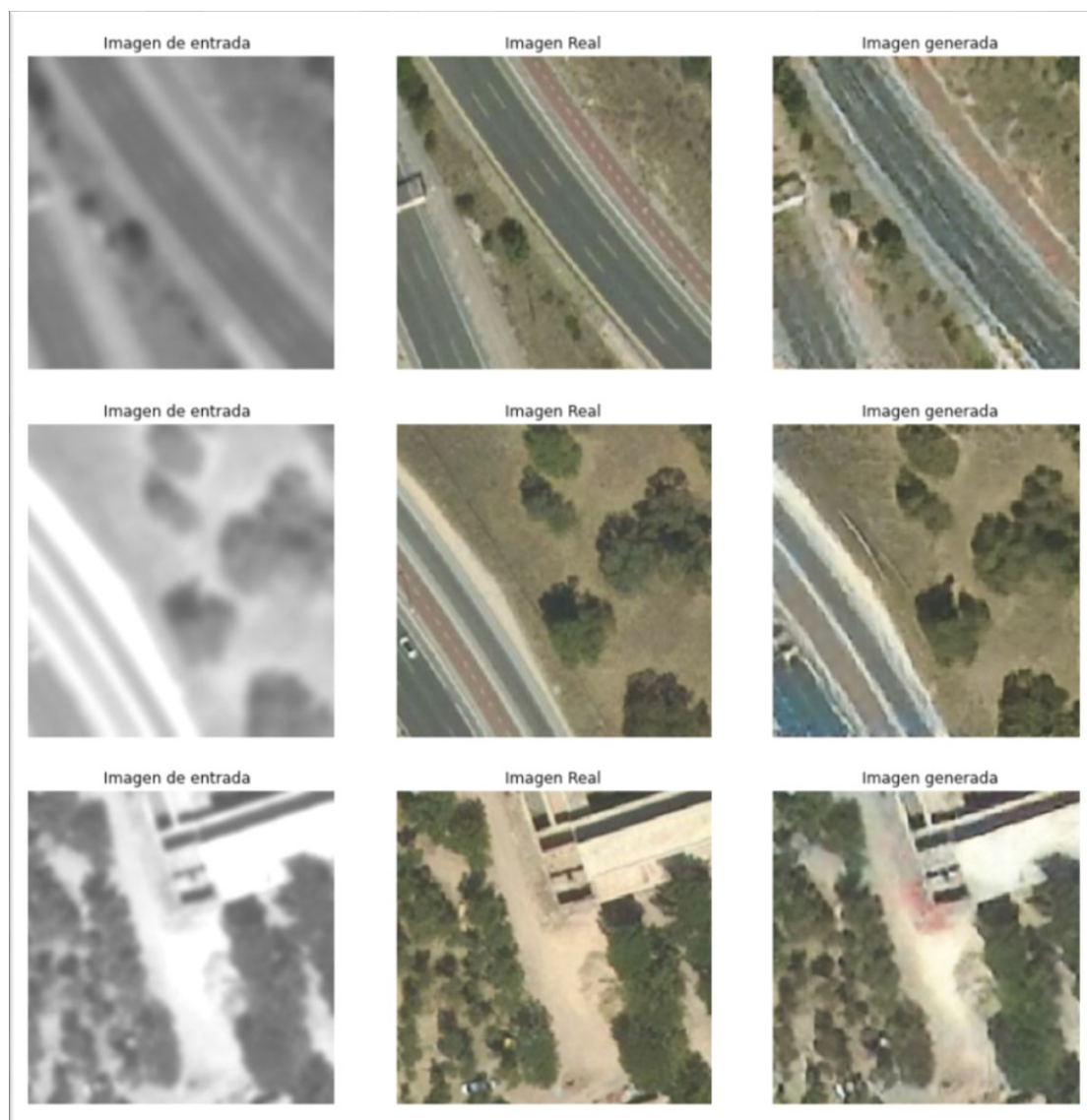


Figura 5.37. Resultados de la aplicación del modelo DeepMapsRestaura19 frente a la imagen original (Imagen Modelo)

Fuente: elaboración propia, a partir de los datos del IGN

En los ejemplos mostrados anteriormente puede apreciarse a través de las imágenes generadas, cómo esta red neuronal es capaz de percibir los elementos y la composición total de la escena presente en la ortofotografía original de forma similar a como lo hace la percepción visual humana y nuestra imaginación para completar aquellos datos faltantes.

6. Generación automatizada de mapas de usos del suelo

En este capítulo, se aborda el desafío de la generación automatizada de mapas de usos del suelo. Para poder mejorar la aplicabilidad de las distintas inteligencias artificiales, se han elaborado varios modelos que reducen el impacto del cambio de dominio usando redes generativas antagónicas condicionales. Estos modelos son los que han sido descritos en los epígrafes anteriores, para la coloración de imágenes, mejora de la resolución, enfoque, eliminación de ruido, etc., todo ello orientado no sólo a mejorar la calidad de los datos disponibles y generar nuevos datos, sino también a adaptar los distintos dominios de las ortoimágenes al de las imágenes del PNOA, con las que han sido entrenados los modelos creados para la segmentación semántica de imágenes; es decir, para su aplicación en la creación automatizada de mapas de usos del suelo. La adaptación del dominio permite adaptar un modelo que ha sido entrenado con un tipo concreto de datos que presentan unas características particulares, a otros datos diferentes; de esta forma se puede, por ejemplo, aplicar una red neuronal que ha sido entrenada para la generación de mapas de usos del suelo a partir de ortoimágenes aéreas del PNOA a los datos procedentes del Vuelo Americano Serie B.

Por otra parte, además de disponer de todas las redes neuronales creadas para la adaptación de dominio, en el proceso de investigación la mayoría de los modelos han sido entrenados para tener una alta tolerancia a imágenes de dominios diferentes; esto ha permitido, entre otras cosas, poder generar mapas temáticos a partir de ortoimágenes de diferentes fuentes. El término dominio puede entenderse de varias formas distintas en lo que a redes neuronales se refiere; puede hablarse del dominio de una función como los distintos valores que pueden adoptar sus variables; también puede utilizarse el término dominio

para determinar el campo de actuación de una red neuronal, por ejemplo asignar colores a imágenes en blanco y negro; pero en este caso utilizamos el término dominio para referirnos a los contenidos de las imágenes utilizadas en un entrenamiento y las características de las mismas. Por otra parte, podemos utilizar el término dominio de forma más restrictiva para referirnos a contenidos concretos de una imagen, como por ejemplo el dominio de densidad forestal del 50% o la identificación del dominio de uso agrícola en la segmentación semántica de las imágenes.

Debido a que en la actualidad existen todavía muy pocos proyectos dedicados a la generación automatizada (sin intervención humana) de mapas de usos de suelos partiendo de ortoimágenes y sobre todo a la naturaleza experimental o meramente teórica de los mismos, no se han podido obtener los modelos ya existentes para realizar pruebas prácticas que permitan la comparación de los resultados que ofrecen con respecto a los obtenidos mediante la aplicación de los creados en esta investigación (ya que su código no es aún público). Hay que añadir que, a pesar de lo mencionado, éste es un campo en constante evolución; de hecho, ni siquiera existían este tipo de aplicaciones cartográficas al comienzo de la elaboración de este documento y, al igual que ocurre con los modelos aquí creados, la precisión de este tipo de redes aplicadas a la segmentación semántica, a través de la cual pueden clasificarse de manera automatizada los datos obtenidos mediante ortofotografía aérea, mejora cada día.

Es importante distinguir entre las tecnologías propuestas para la generación automatizada de mapas de usos del suelo y las tecnologías utilizadas en la actualidad para generar este tipo de mapas. Si bien es cierto que en la actualidad existen varias técnicas de teledetección que facilitan la generación de mapas de usos del suelo, principalmente mediante el uso de ortoimágenes aéreas en la banda pancromática y los canales multispectrales, estas técnicas no son comparables a las aquí propuestas, ya que ni los datos

necesarios son los mismos, ni las métricas de evaluación referidas a la teledetección serían aplicables a la aplicación de redes neuronales para la identificación de elementos concretos y la posterior generación de mapas de uso partiendo de imágenes en tres canales, puesto que la precisión de estas técnicas dependerá siempre de los datos utilizados para alimentar las inteligencias artificiales. En las técnicas existentes y ampliamente utilizadas para la teledetección se recurre a imágenes multispectrales con un número elevado de canales, mientras que las aquí propuestas son eficaces utilizando únicamente los tres canales RGB. Las habituales técnicas de evaluación de la calidad espacial y espectral utilizadas en teledetección, ya sean cualitativas como la medida CIEDE2000 (basada en la evaluación de la distancia entre dos colores en el espacio de color $L^*a^*b^*$, el cual se ha utilizado en los algoritmos de procesamiento de imagen desarrollados en este trabajo y comentados en los capítulos anteriores) o cuantitativas, como las basadas en el concepto de medida de píxeles (ERGAS-IHS, el índice de comparación de frecuencias y la modificación global del índice SSIM), no pueden ser aplicadas a los mapas generados por las inteligencias artificiales elaboradas en esta investigación, debido a que sus resultados están condicionados por los datos con los que han sido entrenadas y, por lo tanto, la evaluación debe realizarse sobre los datos de origen; sin embargo, la eficacia de los modelos creados a la hora de realizar las distintas tareas que se les ha encomendado es fácilmente apreciable de visu. Los únicos métodos de evaluación posible serían los basados en medidas con enfoque de agrupación de píxeles, como la metodología de evaluación basada en objetos, la cual depende de la percepción humana, debiendo hacerse a su vez de visu, por lo que la mejor manera de evaluar los modelos, siempre teniendo en cuenta que sus resultados dependen de la calidad y fidelidad de los datos utilizados para su entrenamiento, sería mediante la revisión visual de los resultados. El hecho de que las inteligencias artificiales creadas tengan una percepción visual similar a la humana y, en muchos casos superior, permitiría su aplicación como sistema de evaluación de técnicas de teledetección basada

en objetos, agilizando este tipo de procesos que hasta ahora han dependido de la intervención humana.

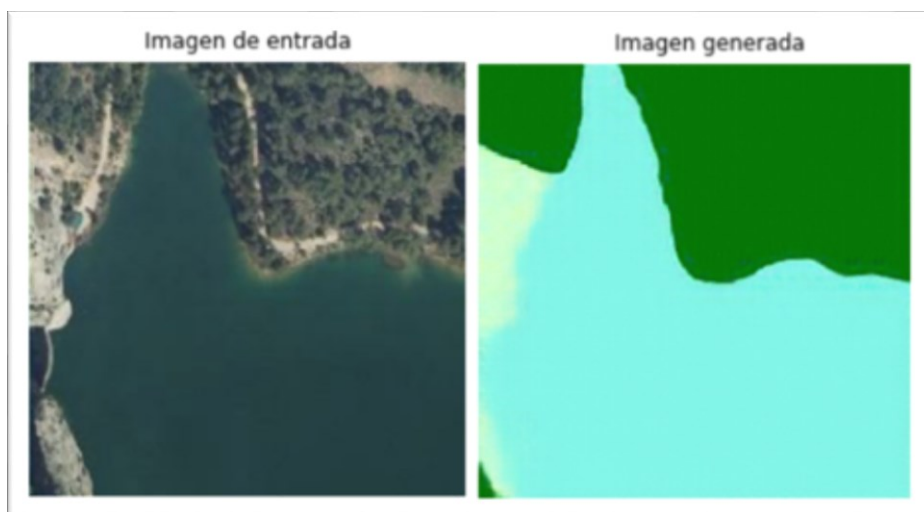


Figura 6.1. Mapa de uso de superficies generado con la inteligencia artificial DeepMapsUsosNatura a partir de ortoimagen del PNOA
Fuente: elaboración propia a partir de datos del IGN

A pesar de que las redes creadas han sido diseñadas para demostrar su posible aplicación práctica, son únicamente una prueba de las posibilidades que ofrece el planteamiento realizado en esta investigación de aplicar redes generativas antagónicas condicionales a trabajos cartográficos. Al igual que las redes propuestas en este capítulo son capaces de realizar la generación automatizada de mapas de usos de suelos a partir de ortoimágenes del PNOA, siguiendo la misma metodología propuesta en esta tesis se podrían crear inteligencias artificiales capaces de transformar imágenes pancromáticas y multiespectrales en mapas de usos de suelos o realizar otras tareas de teledetección, lógicamente estando siempre su precisión limitada a los datos de entrada introducidos. Lo que debe entenderse es que, en cuanto a trabajo cartográfico se refiere, las inteligencias artificiales creadas u otras que se creasen siguiendo la metodología propuesta en este trabajo, tendrán una capacidad de reconocimiento y procesamiento visual que supera incluso las capacidades humanas, por lo que cualquier tarea de teledetección desarrollada por humanos, puede ser superada por estas.

En base a la literatura existente sobre el tema, podemos afirmar que los proyectos que mejores resultados están obteniendo son aquellos basados en las redes generativas antagónicas, es decir, aquellos basados en una arquitectura similar a los modelos desarrollados durante esta investigación. Como ejemplo de ello puede mencionarse la red GeoGAN, la cual, al igual que los modelos aquí propuestos, incluye ruido en el entrenamiento para mejorar la adaptabilidad del modelo.

Los proyectos existentes que utilizan datos cartográficos, al ser en su mayoría pruebas de concepto, han utilizado, casi en su totalidad, los datos de Google Maps, por lo que los mapas generados son similares a éstos. Esta limitación en los datos con los que han sido entrenadas estas redes se debe, probablemente, a que han sido desarrolladas por informáticos y no por cartógrafos; es por ello por lo que se ha recurrido a una fuente de datos cartográficos de uso habitual en la comunidad informática y no a otro tipo de fuentes cartográficas.

En cualquier caso, la segmentación semántica de imágenes aéreas actualmente se centra en zonas urbanas y principalmente para tareas de videovigilancia, basada en la información obtenida mediante drones, la cual no siempre está ortorectificada, por lo que no sirven a propósitos cartográficos. Estos proyectos se ocupan de la tarea de identificar elementos como coches, árboles, personas, etc., pero no son aplicados a la generación de mapas.

Por otra parte, en el caso de los modelos aquí desarrollados, se ha recurrido a datos suministrados por el IDEE y por el IGN, permitiendo así desarrollar las primeras inteligencias artificiales capaces de generar mapas similares a los de la Red Natura 2000, SIOSE o COPENICUS de forma automatizada.



Figura 6.2. Mapa de uso de superficies generado con la inteligencia artificial DeepMapsUsosSIOSE a partir de ortoimagen del Sentinel 2
Fuente: elaboración propia a partir de los datos del IGN

A pesar de que las métricas de evaluación aplicables a los datos generados se basarían en la percepción humana y que los resultados siempre estarán limitados por los datos de origen, tal y como ya se ha comentado, al entrenar las redes aquí propuestas con datos masivos, si la mayor parte de los datos son correctos, la red puede aprender de forma automatizada a corregir errores, las inteligencias artificiales creadas y presentadas en esta tesis son capaces de, incluso, corregir los datos erróneos presentes en su entrenamiento. Esto abre la posibilidad, ya no sólo de generar nuevos datos, sino de, al aplicarlas a los datos ya disponibles, mejorar su calidad y corregir errores. Un buen ejemplo de ello, en cuanto a la generación de mapas de usos del suelo, puede apreciarse en el ejemplo mostrado a continuación.



Figura 6.3. Ortoimagen de entrada procedente del PNOA de Máxima Actualidad
Fuente: elaboración propia a partir de los datos del IGN



Figura 6.4. Mapa de densidad de árboles de Copernicus
Fuente: elaboración propia a partir de los datos del IGN

En el mapa suministrado por Copernicus Land Monitoring Service, puede comprobarse como los métodos de teledetección, basados en el análisis multiespectral, cometen errores como éste, en el cual es evidente que el filtrado de los datos en base a la radiación infrarroja emitida condujo a interpretar como árboles campos de cultivo de cereales, probablemente debido al desarrollo de los cultivos en el momento de la captura de datos. Sin embargo, la inteligencia artificial (DeepMapsBosques), es capaz de evitar este tipo de errores partiendo de menor información que la utilizada en la teledetección de Copernicus, debido a que se basa en el reconocimiento individual de objetos y estructuras, y no en el filtrado de la información recibida por las distintas bandas del espectro. En base a lo anterior, la tecnología propuesta tiene el potencial, no sólo de ofrecer resultados más precisos que las utilizadas hasta ahora, sino también de abaratar los costes de la captura de datos y revolucionar la teledetección ya que los sensores necesarios para obtener los datos de ingesta para el modelo son mucho más económicos que para la obtención de las imágenes pancromáticas y multiespectrales utilizadas hoy en día para el mismo fin.

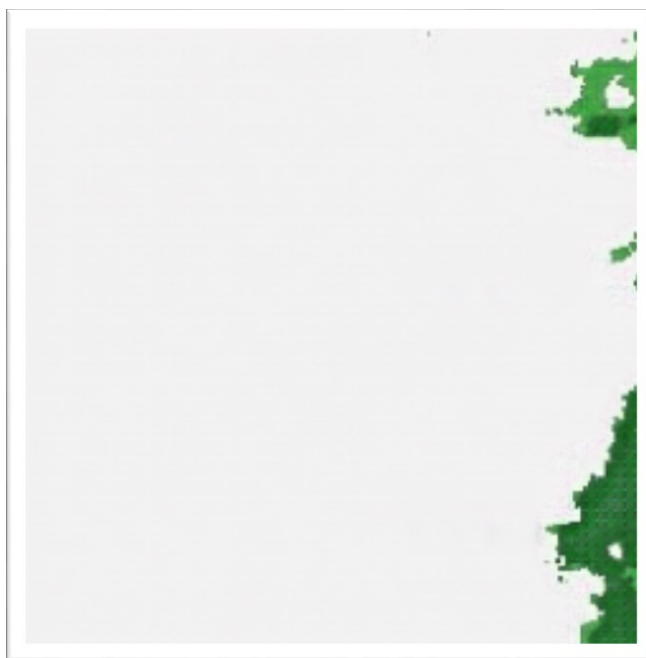


Figura 6.5. Mapa de densidad de árboles generado por el modelo DeepMapsBosques
Fuente: elaboración propia, a partir de los datos del IGN

En la imagen superior puede apreciarse cómo el modelo DeepMapsBosques ha reconocido como árboles únicamente los árboles y no los cultivos de cereales como ocurre con el sistema de teledetección utilizado en el servicio de monitorización del Copernicus.

Tal y como puede comprobarse en el ejemplo anterior los mapas de usos generados mediante la aplicación de la Inteligencia Artificial DeepMapsBosques, permite corregir errores en la teledetección basada en el análisis multiespectral, superando los resultados incluso de las fuentes con las que fue entrenada. De esta forma, se pone de manifiesto cómo la red neuronal DeepMapsBosques produce resultados más veraces que los devueltos por los métodos de teledetección utilizados en Copernicus Land Monitoring Service. Es decir, las únicas técnicas aplicadas en la actualidad para la generación de mapas de usos del suelo que ofrecen un nivel de precisión similar a las inteligencias artificiales creadas en este estudio implican la intervención humana para la revisión de todos los elementos individuales representados en cada uno de los mapas y su edición manual, con el coste que ello conlleva, mientras que los modelos creados para esta investigación son capaces de realizar la misma tarea de forma totalmente automática, sustituyendo la inteligencia humana necesaria hasta ahora por una inteligencia artificial (cuya capacitación es mucho más rápida y económica).



Figura 6.6. Ejemplo de generación automatizada de un mapa de usos del suelo mediante la aplicación del modelo DeepMapsUsosNatura
Fuente: elaboración propia, a partir de los datos del IGN

Por otra parte, cabe mencionar que los modelos aquí creados para la generación automatizada de mapas de usos del suelo son capaces de corregir, no sólo los errores presentes en los mapas actuales derivados de carencias en las técnicas de teledetección, sino también aquellos errores en mapas que han sido elaborados mediante la edición manual realizada por humanos. Este tipo de errores pueden darse por diversos motivos, el siguiente ejemplo muestra cómo el modelo DeepMapsUsosNatura es capaz de corregir los datos de ríos secos categorizados erróneamente como masas de agua debido a la utilización de mapas de ríos ya existentes (y desactualizados) para su elaboración por parte de la RedNatura2000. Tal y como ocurre con el resto de las inteligencias artificiales desarrolladas en este trabajo, DeepMapsUsosNatura es capaz de corregir los errores, no sólo en nuevos datos, sino aquellos que estaban presentes en los datos de entrenamiento, siempre y cuando los datos correctos aportados en el entrenamiento sean suficientes como para que la red aprenda cuál es la tarea encomendada, tal y como se muestra en el siguiente ejemplo.

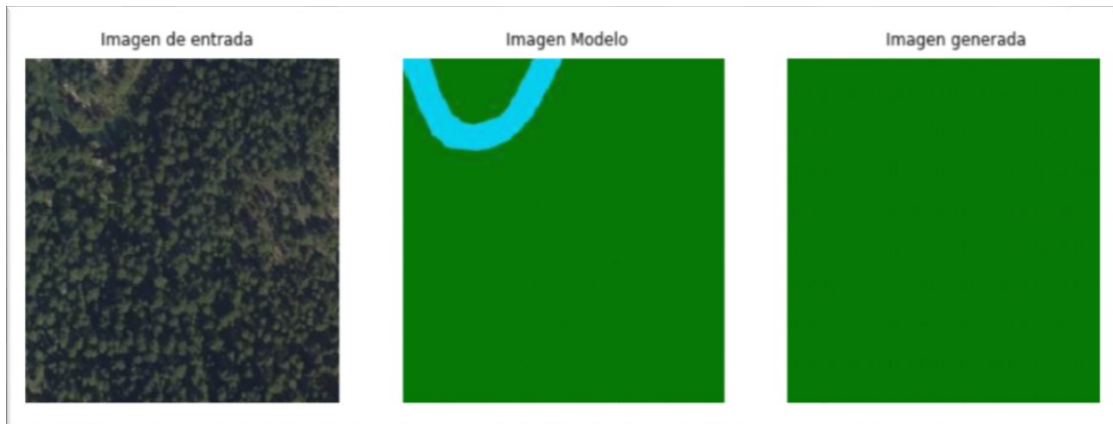


Figura 6.7. Ejemplo de corrección de error de categorización de elementos, presente en los datos de entrenamiento

Fuente: elaboración propia a partir de datos del IGN

Estos errores de categorización derivados del uso de datos desactualizados no son cometidos jamás por las inteligencias artificiales entrenadas durante la presente investigación, ya que la representación de un río o cualquier otro elemento no responde al uso de capas que contengan esa información, sino al análisis visual por parte de una inteligencia de los elementos presentes en una ortoimagen aérea o satelital.

6.1. La adaptación de dominios

La adaptación de dominios es una tarea necesaria si se quiere extender la aplicación de las Redes Neuronales Cartográficas a datos de fuentes distintas a las que se utilizaron en el entrenamiento de estas; es decir, si se entrena una red neuronal con los datos de usos de suelo de la capa Natura 2000 y las imágenes del PNOA de Máxima Actualidad, esta red será capaz de generar imágenes similares a las de Natura 2000 a partir de imágenes del PNOA. Sin embargo, para que se puedan generar este tipo de imágenes partiendo de datos diferentes debe realizarse una fase previa de adaptación del dominio, en la cual se adapte el dominio de origen a aquel con el que fue entrenada la red. Se ha analizado la posibilidad de suprimir esta fase de adaptación de dominio simplificando los datos a partir de los cuales se entrenan las redes que generan mapas de usos del suelo; sin embargo, a pesar de que en la mayor parte de los casos funciona, se produce una pérdida notoria de información durante el proceso de simplificado, lo que implica que la inteligencia artificial se enfrente a un problema a la hora de clasificar algunos elementos, produciéndose así errores. Estos errores, si bien no se eliminan por completo, si pueden minimizarse dividiendo el proceso en dos fases, es decir, una fase inicial de adaptación de dominio, seguida de otra fase de clasificación y generación del mapa de usos del suelo.

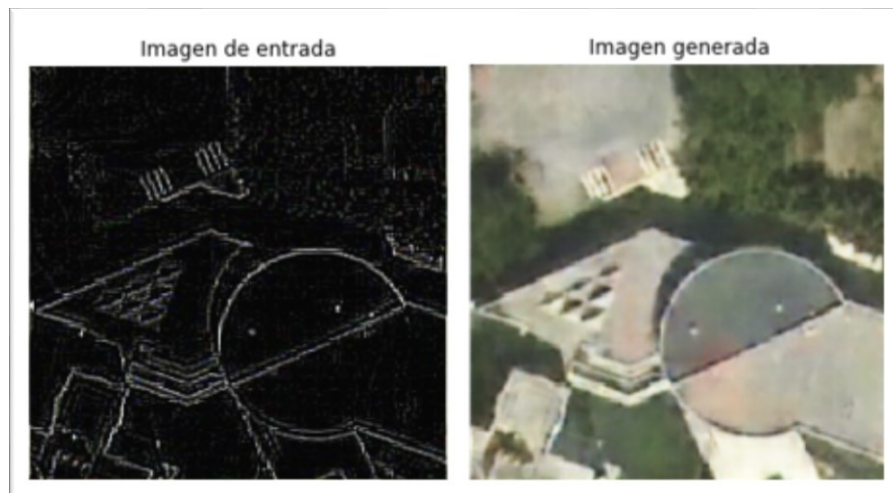


Figura 6.8. Ejemplo de reconstrucción con modelo propio (DeepMapsBordes), de la imagen simplificada

Fuente: elaboración propia, a partir de datos del IGN

La fase de adaptación de dominio puede subdividirse en varios procesos, dependiendo de las diferencias existentes entre el dominio de origen y el de destino. No es lo mismo adaptar el dominio de una ortoimagen aérea en color con alta resolución al dominio del PNOA de Máxima Actualidad que hacerlo con ortoimágenes que tengan casi 100 años, como las del vuelo de Ruiz de Alda. Para poder facilitar esta fase de adaptación y producir herramientas que se puedan utilizar en los procesos necesarios para la misma se han desarrollado las distintas inteligencias artificiales expuestas en este documento, debiendo aplicarse unas u otras dependiendo de la fuente a tratar y teniendo en cuenta que todas ellas están orientadas a normalizar los datos como ortoimágenes aéreas del PNOA de Máxima Actualidad.

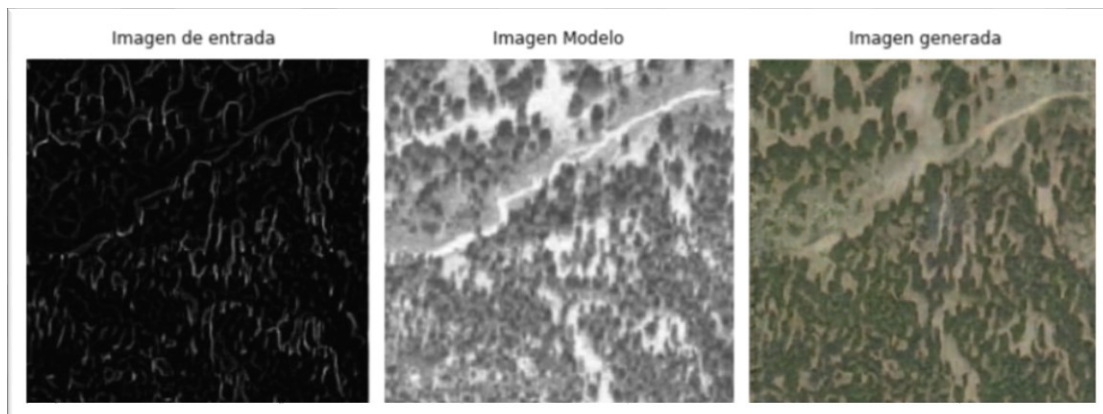


Figura 6.9. Adaptación del dominio del OLISTAT al PNOA con el modelo DeepMapsBordes17
Fuente: elaboración propia a partir de datos del IGN

En base a las conclusiones extraídas de los numerosos modelos entrenados, se han creado Inteligencias Artificiales orientadas específicamente a minimizar el impacto del cambio de dominio. Estos modelos se basan en simplificar la información de las imágenes mediante el preprocesamiento de estas para su posterior recreación mediante Inteligencia Artificial. La simplificación de los datos utilizada se basa en la detección previa de los bordes de una imagen, de esta forma los modelos pueden ser entrenados para reconocer los elementos presentes en las ortoimágenes aéreas a partir, únicamente, de su silueta, tal y como se muestra en la siguiente imagen, en la que un conjunto de árboles ha sido recreado partiendo únicamente de la imagen de los bordes de éstos.

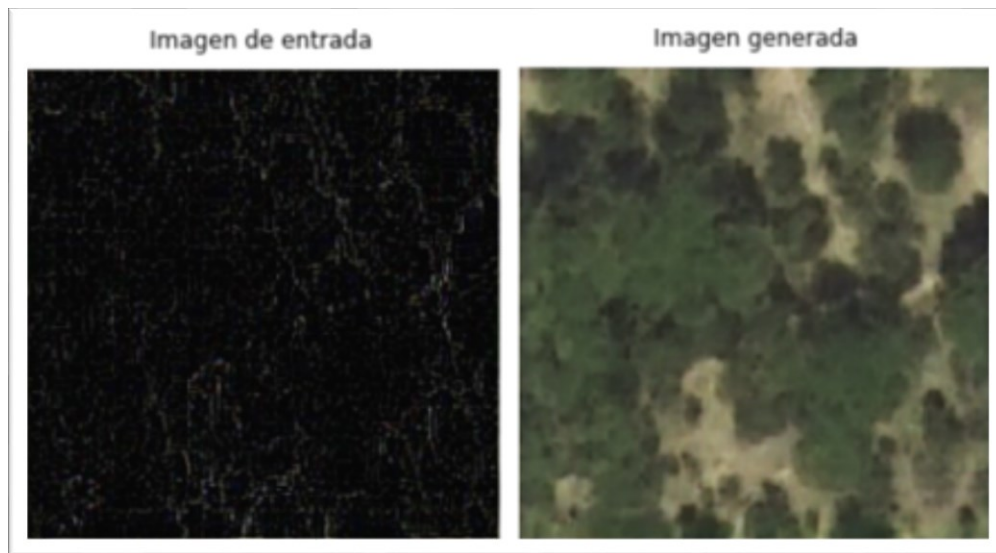


Figura 6.10. Recreación de ortoimagen aérea a partir de imagen de bordes mediante la aplicación del modelo DeepMapsBordes
Fuente: elaboración propia a partir de datos del IGN

Gracias a esto, se limitan los efectos negativos en el cambio de dominio derivados de las particularidades de la imagen como pueda ser su color, resolución, enfoque, etc. Este modelo presenta una serie de ventajas añadidas, pues al simplificar la información se puede editar con mayor facilidad la imagen simplificada que la original; además podría optimizar el almacenamiento de datos cartográficos, ya que estas imágenes simplificadas contienen muchos menos datos que las originales y por lo tanto la memoria que ocupan es mucho menor. Esto permitiría, no sólo un almacenamiento mucho más eficiente de la información cartográfica, sino también su transmisión de forma mucho más rápida. Como puede apreciarse en los ejemplos en los que se ha aplicado este modelo, la imagen regenerada conserva prácticamente toda la información de la imagen inicial.

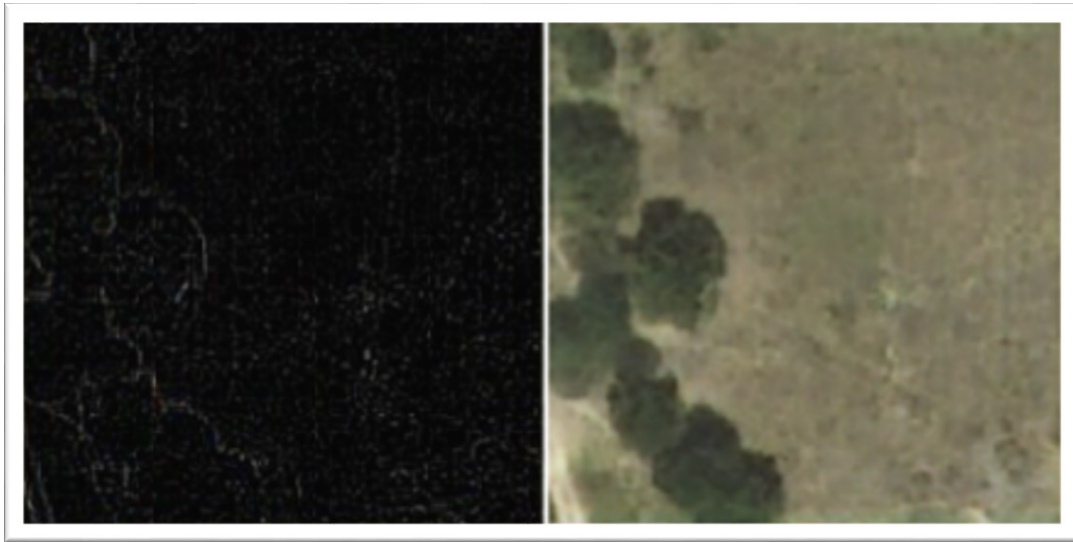


Figura 6.11. Generación de ortoimagen del dominio del PNOA a partir de ortoimagen de bordes

Fuente: elaboración propia, a partir de datos del IGN

Una de las ventajas que se ha podido comprobar al realizar pruebas con los modelos DeepMapsBordes, es que éstos son capaces de generar ortomágenes con las características de las ortofotografías aéreas del PNOA de máxima actualidad, no sólo partiendo de imágenes simplificadas a sus bordes, sino también a partir de ortofotografías en blanco y negro procedentes de fuentes antiguas, pudiendo de esta manera prescindir de la fase de simplificación a la hora de aplicar el modelo a la mayoría de los casos, tal y como se muestra en el siguiente ejemplo, en el cual se ha adaptado el dominio de una ortofotografía aérea de la Calle Alcalá (Madrid) procedente del Vuelo Americano Serie B al PNOA de Máxima Actualidad, mediante el uso de la inteligencia artificial DeepMapsBordes15.



Figura 6.12. Adaptación de dominio del Vuelo Americano Serie B al PNOA de Máxima Actualidad, sin recurrir a la simplificación mediante la generación de bordes (modelo utilizado: DeepMapsBordes15)

Fuente: elaboración propia, a partir de datos del IGN

Aparte de las ventajas que reporta la adaptación de dominios en la aplicación de redes neuronales entrenadas para un dominio determinado a otros dominios, también supone una herramienta muy útil a la hora de completar información faltante en los datos de forma veraz. Como ya se ha comentado en capítulos anteriores, para completar información faltante en una ortoimagen, hasta ahora sólo existían métodos basados en la extrapolación de la información existente en los píxeles circundantes o el uso de parches, lo cual podría mejorar en cierta medida el aspecto, pero estos métodos no proporcionan nueva información verosímil a los píxeles faltantes. Con las redes neuronales propuestas, como DeepMapsInpainting, o cualquiera de las otras redes desarrolladas durante esta investigación, la Inteligencia Artificial es capaz de calcular el contenido más probable en los píxeles faltantes en base al aprendizaje realizado previamente; por ejemplo, si falta una fracción de un bosque las inteligencias aquí desarrolladas son capaces de completar esa fracción con árboles similares al resto de los existentes en ese bosque. Si bien la información añadida es mucho más plausible y verosímil que la aportada por

métodos anteriores, esta información no tiene por qué ser veraz, ya que en esa fracción desaparecida de la imagen podía existir un elemento anómalo que ni una inteligencia artificial ni una humana podrían predecir. Los modelos de inteligencias artificiales aquí creadas para la adaptación de dominios pueden aportar datos veraces para sustituir los datos faltantes, ya que a través de la adaptación del dominio de datos adquiridos en fechas cercanas a la captura de los datos que se pretende completar puede adquirirse la información de los elementos presentes en el área sin información y posteriormente realizar una recreación verosímil de dichos elementos.

Un ejemplo de esta adaptación de dominios para reparar datos faltantes a partir de datos de otras fuentes, obtenidos en fechas similares sería la Inteligencia Artificial DeepMapsHillshades, a través de la cual pueden recrearse imágenes similares a los hillshades derivados de los datos LiDAR a partir de ortoimágenes aéreas. Para aumentar la versatilidad de esta inteligencia, se han reducido los datos del PNOA con los que ha sido entrenada a escala de grises, de esta forma podrían generarse hillshades partiendo de ortoimágenes aéreas de fuentes más antiguas que el propio PNOA.

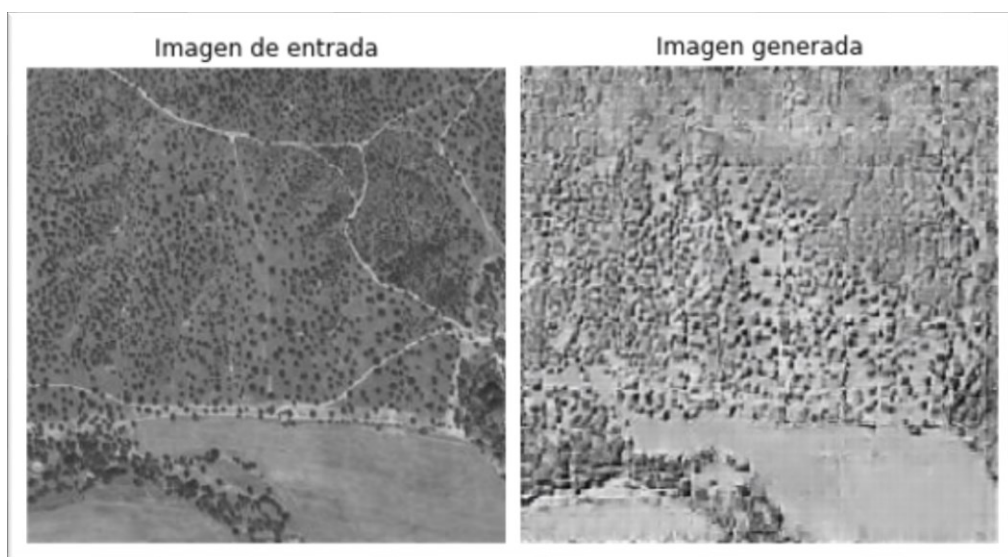


Figura 6.13. Ejemplo de transformación de ortoimágenes aéreas a hillshades mediante el uso de la inteligencia artificial DeepMapsHillshades
Fuente: elaboración propia, a partir de datos del IGN

De todas formas, para solucionar los defectos en datos procedentes del LiDAR, la mejor solución es el correcto tratamiento de las nubes de puntos originales, realizando un correcto filtrado de los datos y ampliando la información faltante mediante técnicas tridimensionales como las expuestas en los capítulos 8 y 9, lo cual resulta sencillo utilizando el software y metodologías creados durante esta investigación. La Inteligencia Artificial DeepMapsHillshades, ha sido creada para poder generar hillshades en los casos en los que no se dispone de la nube de puntos y sólo se tienen ortoimágenes aéreas.

6.2. Modelos propios

Aunque muchas de las inteligencias desarrolladas en esta investigación pueden aplicarse a la adaptación de los dominios de distintas ortoimágenes, tienen otras aplicaciones, como la restauración de ortoimágenes dañadas, la adición de color a ortoimágenes antiguas en blanco y negro, la mejora de la resolución o el enfoque, etc. Sin embargo, también se han desarrollado modelos específicamente para realizar esta adaptación de dominio. Los modelos expuestos a continuación son un ejemplo de algunas de estas inteligencias artificiales, que tienen como objetivo adaptar una fuente de ortoimágenes a otra.

6.2.1. DeepMapsBordes

Este modelo ha sido entrenado para permitir la adaptación de dominios de ortoimágenes, con diferente origen, mediante su simplificación a una representación de los bordes. Para su entrenamiento han sido utilizadas 4.386 teselas de 256x256 píxeles, seleccionadas de forma aleatoria, procedentes de las hojas del PNOA de Máxima Actualidad, correspondientes a la Comunidad de Madrid, y con un nivel de zoom de 19, equivalente a una escala de 1:1.000. Este modelo ha sido entrenado durante 2.000 iteraciones. Para generar las imágenes con las que se ha entrenado el modelo, se ha realizado un procesado de las imágenes mediante un script de Python desarrollado para tal fin, lo que ha permitido generar el par de imágenes necesario, constituido por una imagen de los bordes y la imagen original del PNOA, tal y como se muestra a continuación.

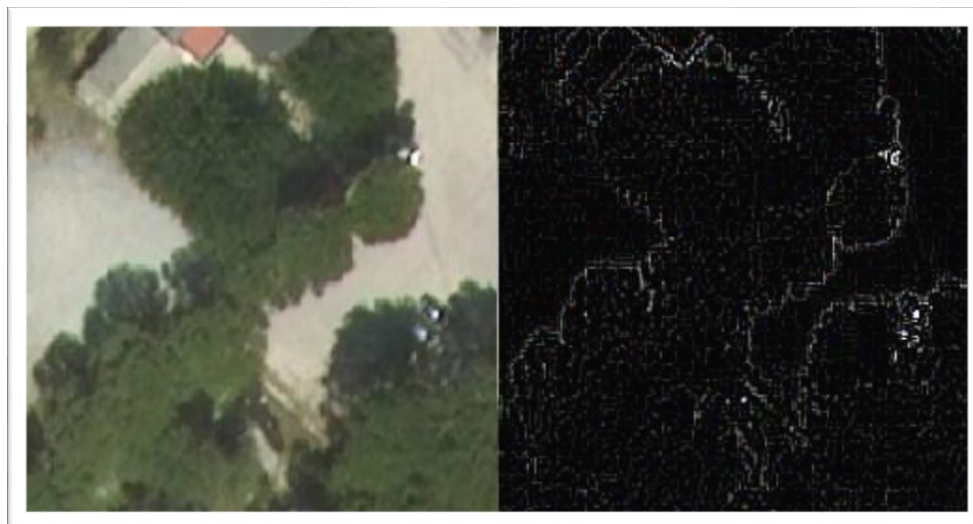
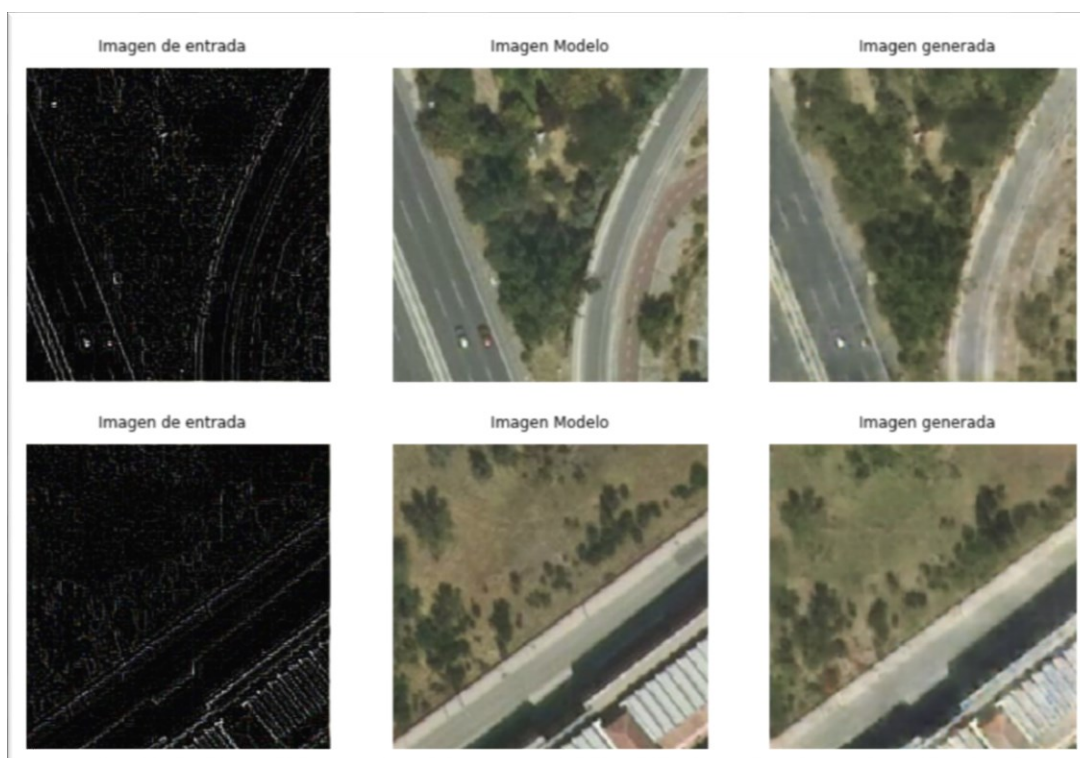


Figura 6.14. Ejemplo de par de imágenes generado para el entrenamiento del modelo DeepMapsBordes

Fuente: elaboración propia, a partir de los datos del IGN

Para generar las imágenes que se muestran como ejemplo, a continuación, se han escogido teselas pertenecientes a la hoja 0534, correspondientes a la Universidad Autónoma de Madrid y áreas colindantes.



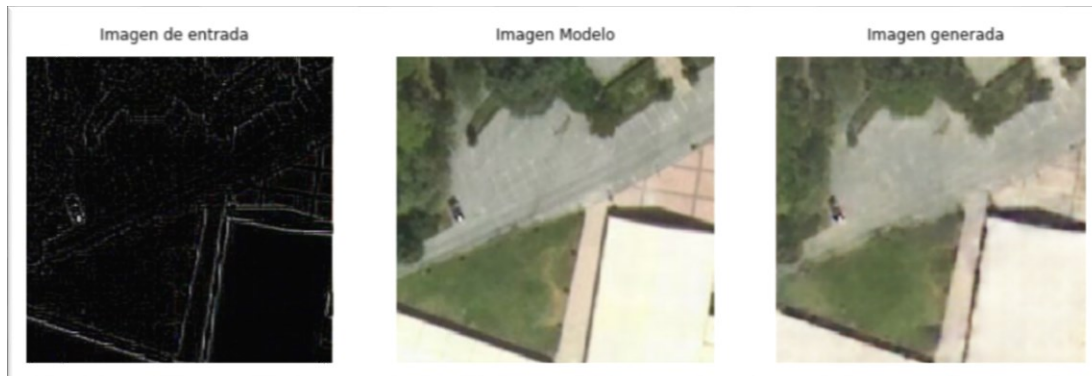
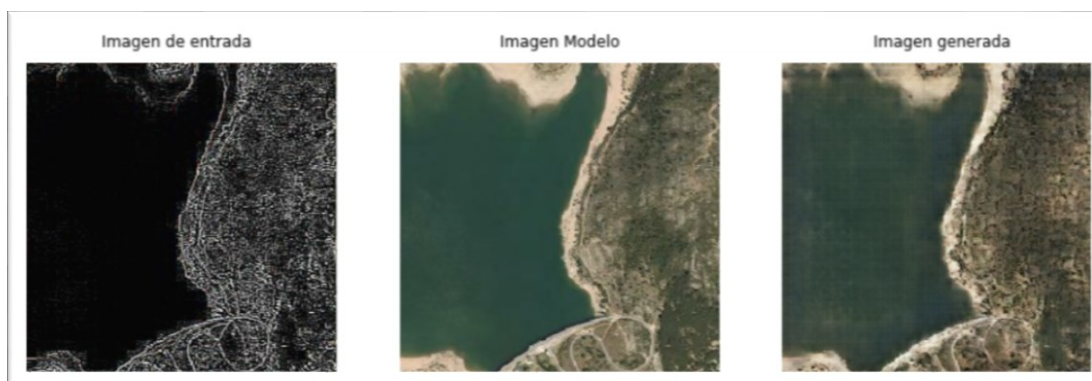


Figura 6.15. Resultados de la aplicación del modelo DeepMapsBordes
Fuente: elaboración propia, a partir de los datos del IGN

6.2.2. DeepMapsBordes15

Para el entrenamiento de este modelo se ha partido de los datos de las hojas del PNOA de Máxima Actualidad, correspondientes a Castilla-León, la Comunidad de Madrid y Castilla la Mancha, de las cuales se han seleccionado de forma aleatoria 5.374 teselas de 256x256 píxeles. A partir de estos datos iniciales, se ha realizado un procesamiento de las imágenes mediante un script de Python para generar los pares de imágenes necesarios para el entrenamiento, los cuales constaban de una imagen simplificada en la que se representan únicamente los bordes y la imagen original del PNOA. El entrenamiento constó de 6.000 iteraciones y dio los siguientes resultados.



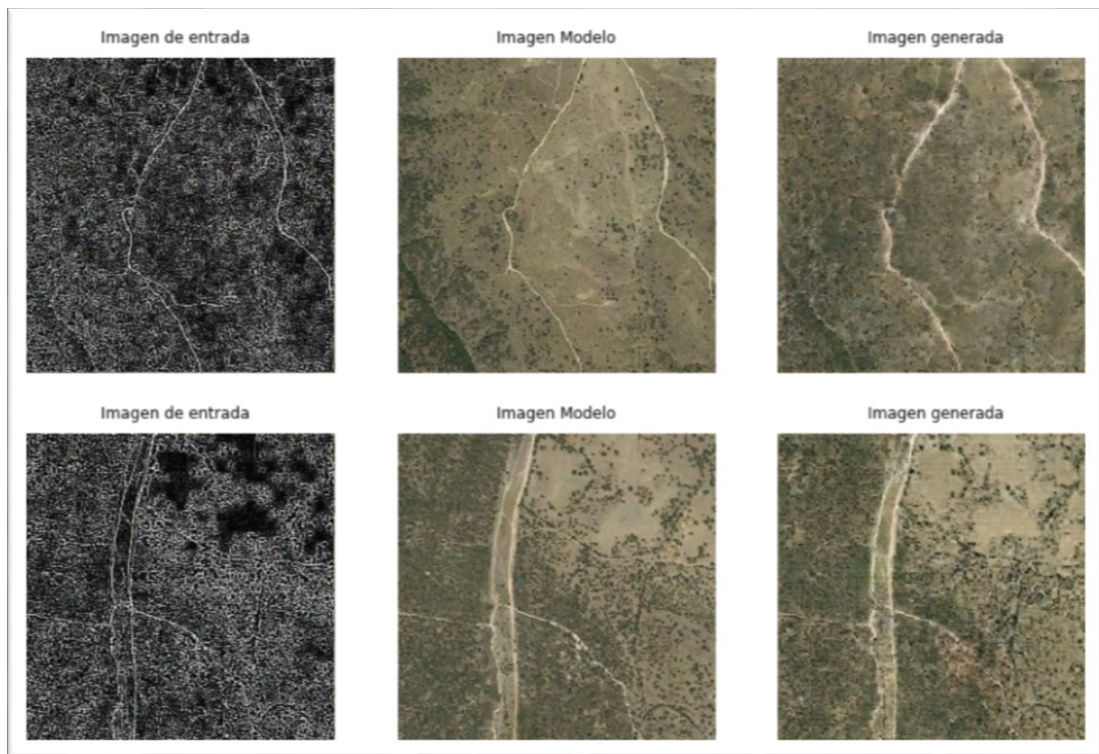


Figura 6.16. Resultados generados por el modelo DeepMapsBordes15
Fuente: elaboración propia a partir de datos del IGN

6.2.3. DeepMapsBordes17

Esta inteligencia ha sido entrenada con ortoimágenes aéreas procedentes del PNOA de Máxima Actualidad y con los datos derivados de ellas al aplicar el script de Python diseñado para simplificar su información. A diferencia de los modelos mencionados anteriormente, en este caso se ha mejorado la versatilidad del modelo añadiendo ruido, eliminando detalles de los datos de entrenamiento, generando varias resoluciones y juntando estos datos con los datos originales mediante un script de Python creado expresamente para ello. A continuación, se muestran ejemplos de los datos de entrenamiento utilizados.

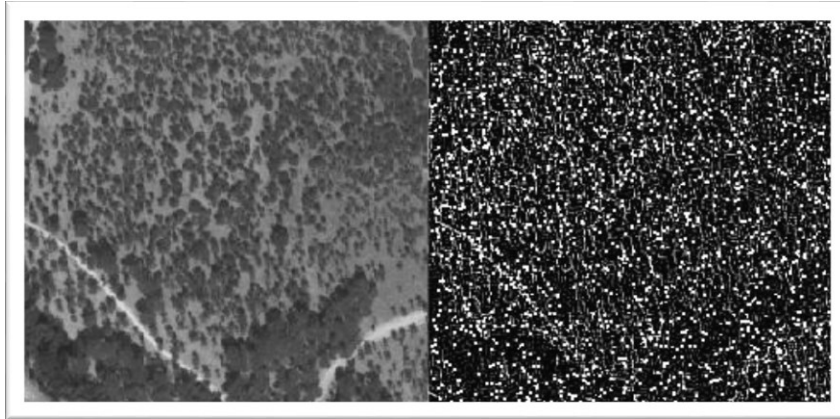


Figura 6.17. Ejemplo de par de imágenes utilizadas en el entrenamiento en el que se han incluido datos simplificados con ruido
Fuente: elaboración propia, a partir de datos del IGN

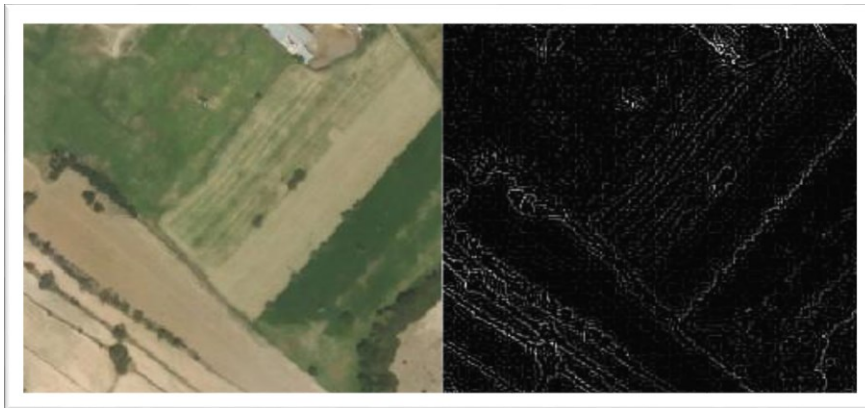


Figura 6.18. Ejemplo de par de imágenes utilizadas en el entrenamiento en el que se han incluido datos simplificados sin ruido
Fuente: elaboración propia, a partir de datos del IGN

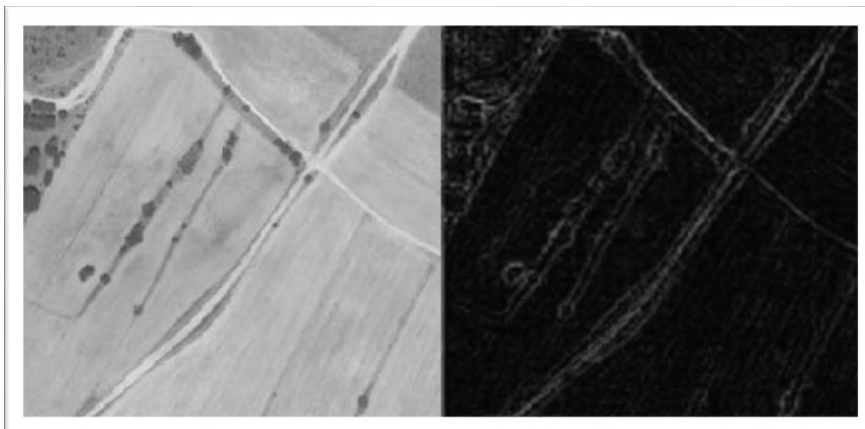


Figura 6.19. Ejemplo de par de imágenes utilizadas en el entrenamiento en el que se han incluido datos reescalados y simplificados sin ruido
Fuente: elaboración propia, a partir de datos del IGN

Los datos han sido obtenidos de teselas de 256x256 píxeles con un nivel de zoom de 17, equivalente a una escala aproximada de 1:4.000, seleccionadas de forma aleatoria a partir de las hojas correspondientes a la Comunidad de Madrid y a Castilla la Mancha. El modelo ha sido entrenado con 2.532 imágenes durante 8.100 iteraciones. Gracias al preprocesado de los datos de entrenamiento, este modelo acepta ortoimágenes de prácticamente todas las fuentes.

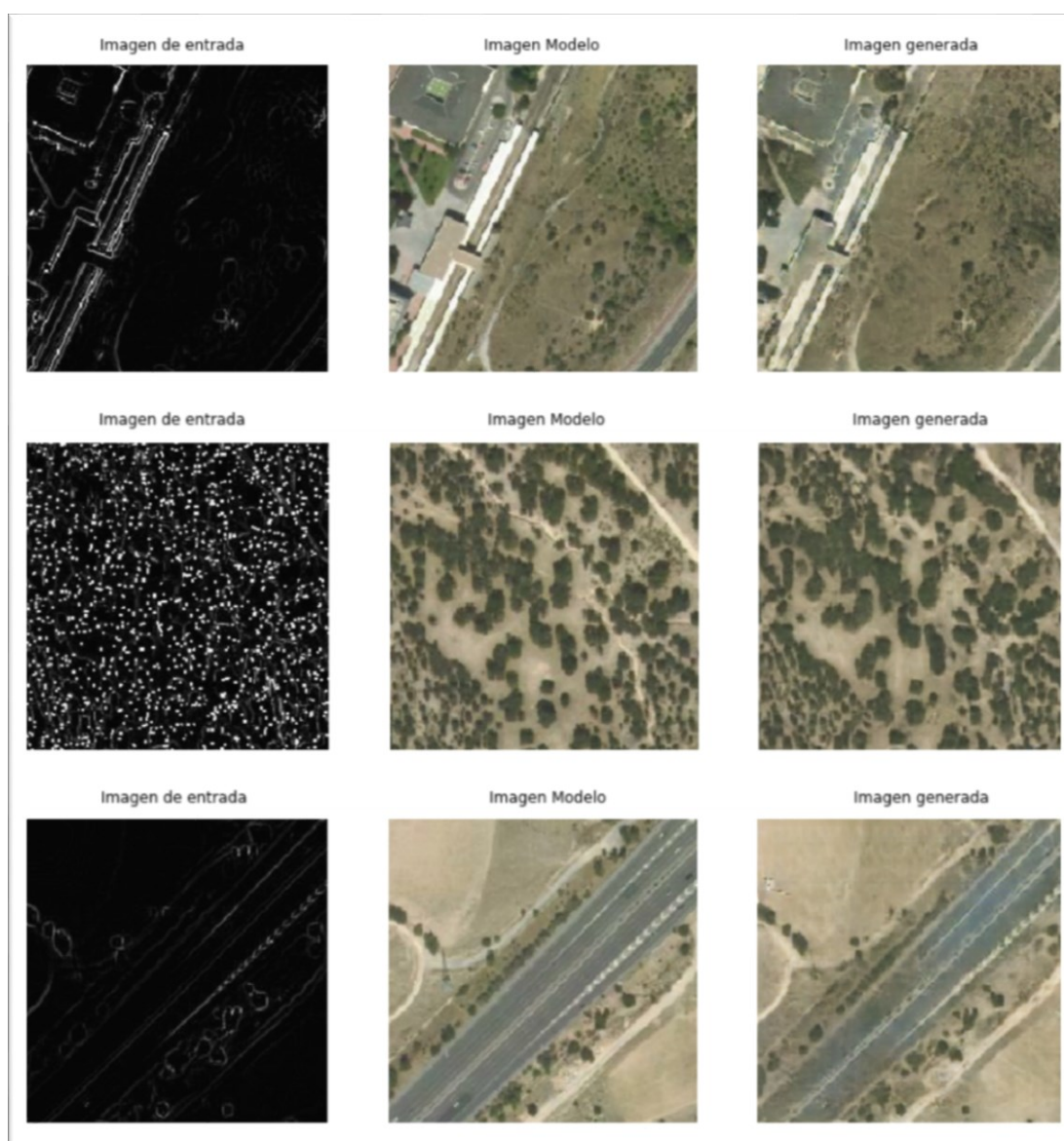
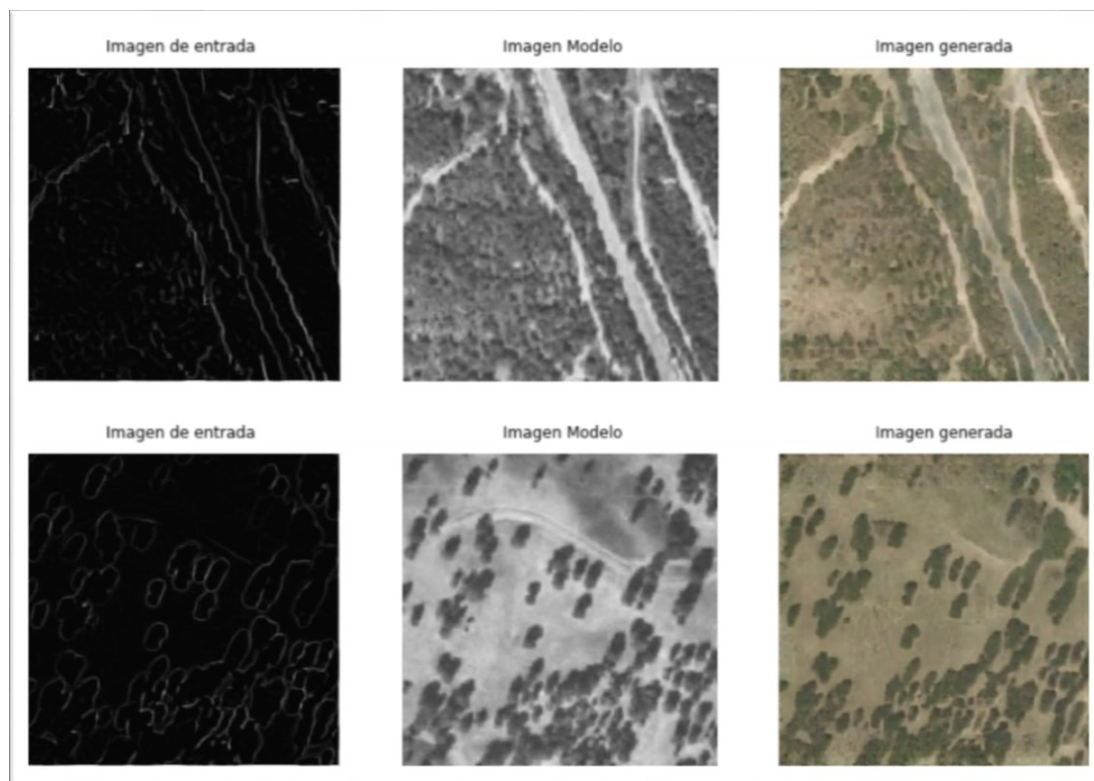


Figura 6.20. Resultados de la aplicación del modelo DeepMapsBordes17
Fuente: elaboración propia, a partir de datos del IGN

A continuación, se puede ver como los modelos entrenados para la adaptación de dominios son capaces de transformar la información de un dominio a otro.

En este caso, transforma la información contenida en las capas de OLISTAT (en escala de grises) a PNOA (en color). Para ello, como ya se ha mencionado en este capítulo, se realiza un preprocesamiento de las imágenes para obtener una imagen simplificada con los bordes de la imagen original, que es la que se ingesta en el modelo y finalmente da lugar a una ortoimagen con las características propias de las procedentes del PNOA.



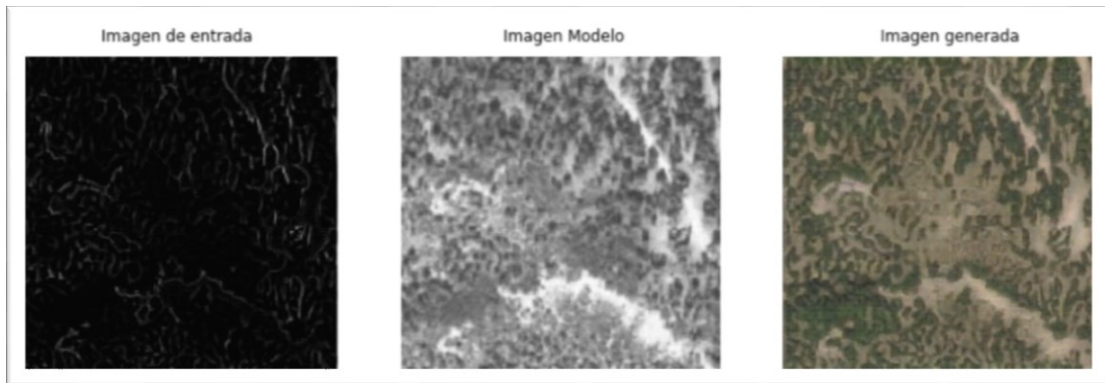


Figura 6.21. Resultados de la aplicación del modelo DeepMapsBordes17 para adaptar el dominio de ortoimágenes del OLISTAT

Fuente: elaboración propia, a partir de datos del IGN

Como puede observarse en las imágenes anteriores, las redes entrenadas para adaptar dominios producen como consecuencia, la coloración automatizada de aquellos dominios que originariamente se encontraban en escala de grises.

6.2.4. DeepMapsHillshades

Este modelo ha sido entrenado a partir de Ortoimágenes aéreas del PNOA de Máxima Actualidad y de los datos del Modelo Digital de Superficies LiDAR de España ofrecido por el IDEE correspondientes a todo el territorio nacional.

Las teselas correspondientes al PNOA de Máxima Actualidad han sido generadas a partir de la hoja MTN50 correspondientes a todo el territorio nacional descargadas del IGN, mientras que las teselas relativas al Modelo Digital de Superficies LiDAR se han generado a partir de los datos obtenidos mediante el siguiente servicio WMTS <http://wmts-mapa-LiDAR.idee.es/LiDAR?request=GetCapabilities&service=WMTS&>”.

Todas las teselas generadas, han sido creadas con un nivel de zoom de 15, es decir, con una escala aproximada de 1:15.000.

El modelo ha sido entrenado con 3.846 pares de imágenes como el mostrado a continuación, durante un total de 6.200 iteraciones.

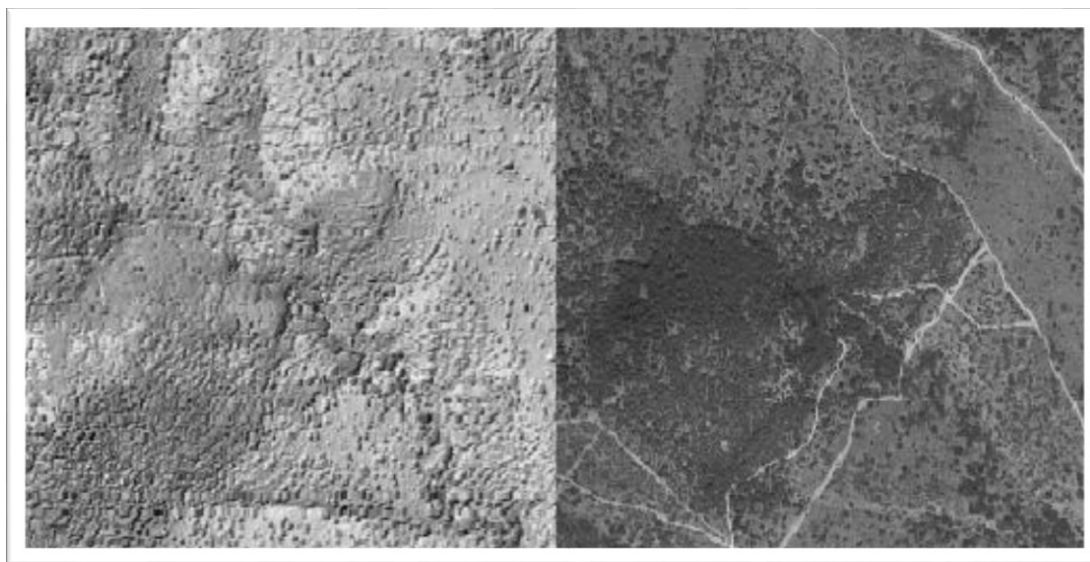


Figura 6.22. Ejemplo de par de imágenes utilizado en el entrenamiento del modelo
DeepMapsHillshades

Fuente: elaboración propia a partir de datos del IGN

El Modelo Digital de Superficies LiDAR ha sido generado por composición de cuatro capas: sombreado, edificaciones, vegetación e hidrografía. A partir de las nubes de puntos LiDAR, correspondientes a la primera cobertura del proyecto PNOA-LiDAR, debidamente clasificadas, se genera un modelo digital de superficies al que se le aplica un sombreado. Sobre este sombreado se superponen tres capas: dos modelos digitales de superficies normalizados, uno de la clase vegetación y otro de la clase edificación, y la rasterización de la capa de hidrografía de la Información Geográfica de Referencia del Sistema Cartográfico Nacional.

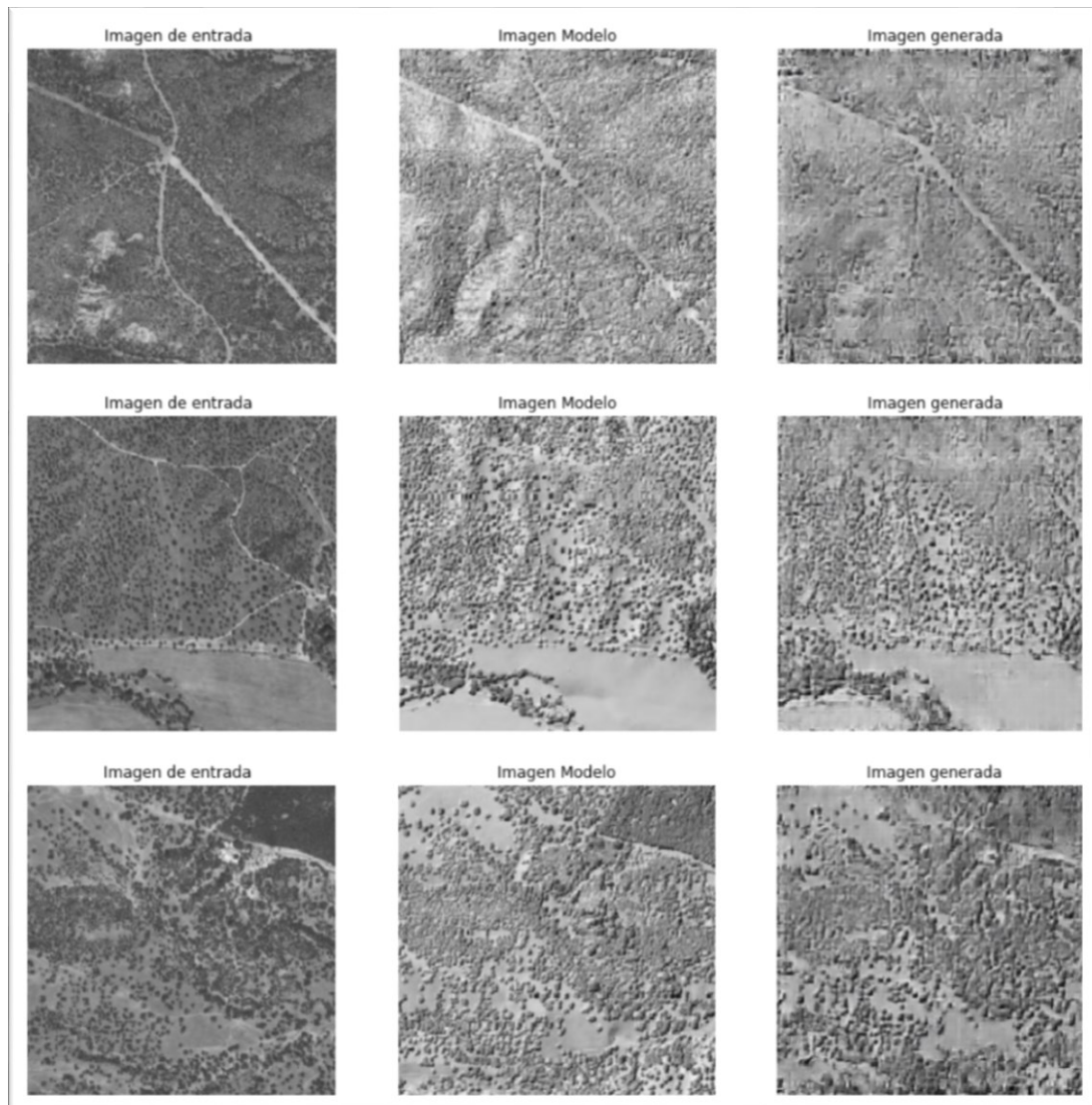


Figura 6.23. Resultados del entrenamiento de DeepMaps Hillshades en relación con las imágenes utilizadas como modelo

Fuente: elaboración propia a partir de datos del IGN

6.3. La segmentación semántica

Para poder acometer la tarea de generar mapas de usos del suelo, las inteligencias artificiales deben ser capaces de realizar un reconocimiento de los elementos presentes en las mismas y asociarlos a distintos conceptos, es decir, cuando vean un árbol deben saber que se trata de un árbol para después asignarle un color y construir el mapa.

Este proceso de identificación de elementos se conoce como segmentación semántica. Tradicionalmente, la segmentación semántica se considera una tarea de análisis de imagen, que asigna para cada píxel de una imagen de entrada una etiqueta.

La novedad que se propone, para que la generación de mapas sea directa, es el hecho de optimizar el proceso mediante la sustitución de esa etiqueta por colores de píxeles, es decir, en lugar de asociar los píxeles de una ortofotografía correspondientes a un bosque a la etiqueta bosque, éstos se asocian a otros píxeles del color correspondiente con los bosques en la leyenda del mapa de usos de suelo que se va a generar. Más allá de la detección de objetos y la clasificación de la imagen, la segmentación semántica es la tarea de análisis de imágenes de más alto nivel que permite una completa comprensión de la escena de toda la imagen de entrada. La segmentación semántica fue referida en muchos trabajos de teledetección como clasificación por píxeles.

El término de segmentación semántica es más usado en la visión computarizada y está siendo cada vez más adoptado en la teledetección. Esta técnica se puede utilizar en las imágenes aéreas en una variedad de posibles aplicaciones, como la vigilancia y la planificación de las zonas urbanas, la gestión y el análisis del tráfico, áreas en las que se han focalizado los distintos trabajos realizados hasta ahora, dejando a un lado tareas clásicas de la

teledetección como la generación de mapas de usos del suelo a partir de ortofotografías. Este tipo de usos se ha visto potenciado por el desarrollo e implantación de los drones.

El progreso de los algoritmos de segmentación semántica se retrasó hace años por la baja precisión de los enfoques tradicionales de los algoritmos de análisis de imágenes, basados en la extracción de características de forma manual. Pero desde la aparición de extractores de rasgos altamente descriptivos, como las Redes Neuronales Convolucionales, toda el área de análisis de imágenes ha registrado un aumento significativo de la precisión. De hecho, en esta última década dichas redes han demostrado ser la herramienta más eficaz en la visión computarizada.

Aunque la arquitectura de las Redes Neuronales Convolucionales se introdujo por primera vez en 1998, la adopción de este tipo de redes se vio inhibida por los pesados cálculos necesarios para que estas pudiesen ser entrenadas para labores de visión computarizada. Sin embargo, este tema se ha resuelto recientemente con las mejoras realizadas en las Unidades de Procesamiento Gráfico (GPU). Éste es el motivo por el que, al no disponer localmente de los recursos técnicos para entrenar las redes, se ha optado por utilizar procesamiento distribuido, mediante el uso de los servidores de Google Colaboratory. Una de las mejoras surgidas con las GPU es la posibilidad de paralelización de los cálculos de entrenamiento sobre los múltiples núcleos de la GPU, reduciendo así el tiempo de entrenamiento de las redes neuronales y promoviendo la adopción de dicho modelo, de forma práctica, para el análisis de imágenes. El éxito de estas arquitecturas puede explicarse por el hecho de que las Redes Neuronales Convolucionales son capaces de extraer múltiples niveles de representación para de esta forma aislar los objetos de interés. Estas representaciones se entrenan a través de las diferentes capas a fin de ser invariables a la luz, a la escala y a la rotación.

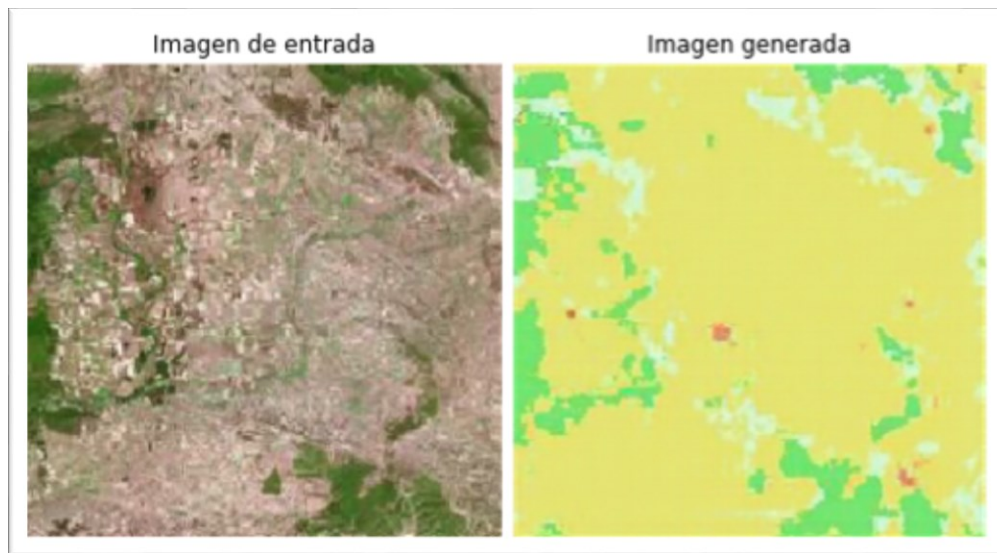


Figura 6.24. Ejemplo de mapa de usos de suelo generado por el modelo de creación propia
DeepMapsUsosSIOSE10
Fuente: elaboración propia, a partir de los datos del IGN

6.4. Modelos propios

En este epígrafe se exponen algunas de las inteligencias artificiales creadas durante esta investigación, que tienen como objetivo la creación automatizada de mapas de usos del suelo, para el entrenamiento de las mismas se han utilizado distintas fuentes de datos de mapas de usos del suelo, como demostración de la capacidad de los modelos basados en la arquitectura propuesta para la creación automatizada de mapas, que presentan en origen distintas características y mostrando, a su vez, como se pueden adaptar a distintas tareas de generación automatizada de mapas de tipos diversos, incluyendo números variables de elementos a representar (pueden representarse mapas que tengan más o menos categorías).

Para demostrar la capacidad de las inteligencias artificiales a la hora de generar mapas de usos en distintas escalas, además del modelo DeepMapsUsosNatura que ha sido creado para generar mapas de usos precisos a escalas cercanas a 1:4.000, se han entrenado el modelo DeepMapsUsosSIOSE, el cual está orientado a escalas menores cercanas a 1:70.000, y el modelo DeepMapsSuperficies para escalas cercanas a 1:500.000.

6.4.1. DeepMapsUsosNatura

Esta Inteligencia Artificial ha sido entrenada para poder generar mapas de usos del suelo similares a los proporcionados por el IGN, en base a los datos de la Red Natura 2000. Para su entrenamiento se han utilizado teselas de 256x256 generadas a partir, tanto de los datos del PNOA de Máxima Actualidad como de la Red Natura, de las hojas correspondientes a la Comunidad de Madrid. Este modelo ha sido entrenado con 1.024 imágenes durante 2.000 iteraciones.

Las imágenes de los mapas generados por el modelo DeepMapsUsosNatura se corresponden con la siguiente leyenda:

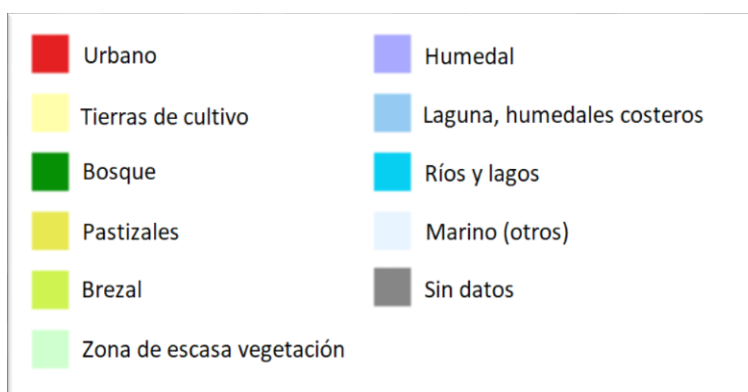


Figura 6.25. Leyenda de los mapas de uso generados por la inteligencia artificial DeepMapsUsosNatura y de los usados como entrenamiento
Fuente: elaboración propia, a partir de los datos de Natura 2000

Debe destacarse el hecho de que, en los resultados obtenidos, se aprecia cómo este modelo es capaz de clasificar los elementos presentes en las ortoimágenes y colorear los píxeles correspondientes, en base a la leyenda de la Red Natura 2000; sin embargo, determinados elementos no apreciables mediante la inspección visual, en las ortoimágenes del PNOA, no los representa, como ocurre con los ríos que aparecen secos en los datos del PNOA. Esta aparente limitación constituye a la vez una ventaja, ya que se ha podido comprobar que en muchos casos esta Inteligencia Artificial es capaz de corregir datos erróneos y de generar mapas de usos del suelo con más detalle que los originales. El hecho de que pueda corregir datos erróneos se debe a que la inmensa mayoría de las correspondencias entre píxeles del PNOA y de la Red Natura 2000 era correcta, por lo que el modelo ha aprendido a generar los nuevos datos de forma precisa, incluso para aquellos datos cuya correspondencia era errónea y fueron utilizados para el entrenamiento. Para ilustrar este hecho, a continuación, se incluye una muestra de las ortoimágenes generadas.

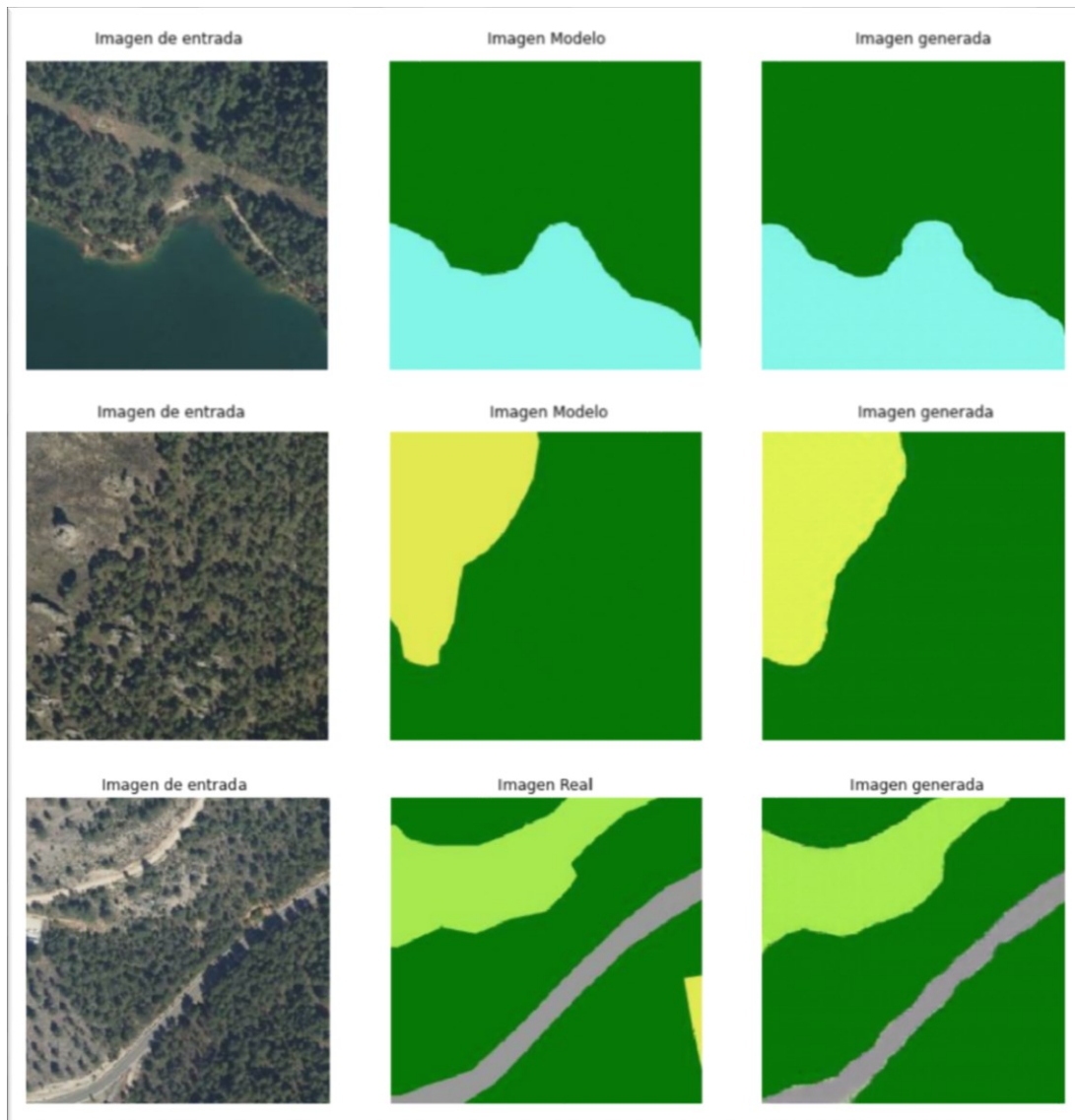


Figura 6.26. Resultados de la aplicación del modelo DeepMapsUsosNatura a Ortofotografías aéreas del PNOA

Fuente: elaboración propia, a partir de datos del IGN

Una de las ventajas que tiene esta Inteligencia Artificial es que puede generar mapas de usos de superficies cada vez que se disponga de nuevas ortofotografías. A continuación, se muestran dos ejemplos en los que se puede comprobar como la Red Neuronal mejora, incluso, los datos con los que ha sido entrenada, generando mapas de usos del suelo más cercanos a la realidad que los originales con los que fue entrenada.

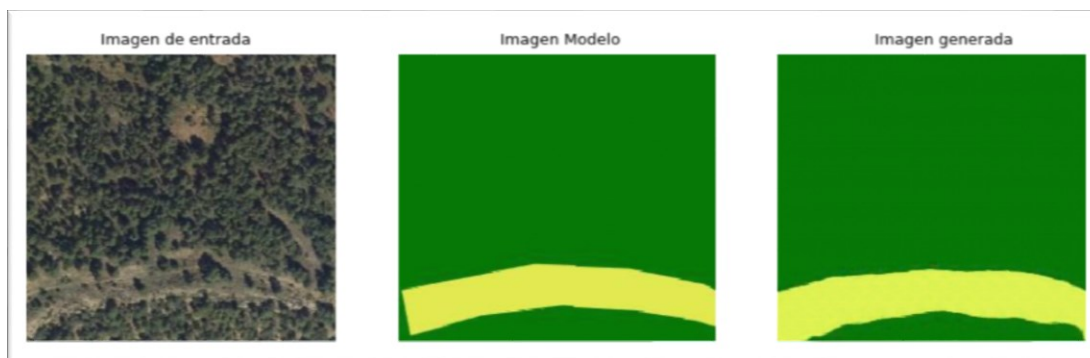


Figura 6.27. Ejemplo de mejora del mapa de usos original mediante la utilización de la Inteligencia DeepMapsUsosNatura
Fuente: elaboración propia, a partir de datos del IGN

En este ejemplo se ve como la imagen original (Imagen Modelo), no se aproxima tanto a la realidad representada en la ortofotografía utilizada como imagen de entrada como la imagen generada, la cual no sólo se ajusta mejor, sino que representa la realidad más fielmente, añadiendo cierto tono verde a la categoría de pastizal para representar la presencia de algunos árboles dispersos en el pastizal.



Figura 6.28. Ejemplo de la actualización de datos del mapa de usos original mediante la utilización de la Inteligencia Artificial DeepMapsUsosNatura
Fuente: elaboración propia, a partir de datos del IGN

En la imagen superior puede verse cómo la Inteligencia Artificial DeepMapsUsosNatura es capaz de actualizar los datos existentes, ya que la ortoimagen generada no recoge como curso de agua un río que ya está seco y las zonas de bosque y de prado se aproximan más a la realidad que en el mapa

de usos de superficies de Natura2000; además, se aprecia como el color asignado a los pastizales es algo más verde que el original. La asignación de una tonalidad más verdosa es consecuencia de que, debido a la existencia de algunos árboles y arbustos, el modelo generado ha creado una categoría híbrida entre los pastizales, los brezales y los bosques (predominando el color de los pastizales sobre los otros, ya que el porcentaje de superficie ocupada en esa zona por los pastizales es mayor) para representar un pastizal en el que hay algunos arbustos. Esta nueva tonalidad no presente en la leyenda original muestra la capacidad de las redes desarrolladas en este estudio para crear nuevas categorías a partir de las existentes en los datos de entrenamiento, siendo este fenómeno un buen ejemplo de autoemergencia de un nuevo comportamiento para el que la red no ha sido entrenada. Estas nuevas categorías no suponen un problema, ya que si se quiere restringir las categorías en los datos generados a las de los datos de entrenamiento, podría hacerse mediante un sencillo postprocesamiento de las imágenes generadas.

6.4.2. DeepMapsBosques

El presente modelo ha sido entrenado con los datos de Copernicus Land Monitoring Service, obtenidos a través del servicio WMS “<https://servicios.idee.es/wms/copernicus-landservice-spain?SERVICE=WMS&>”. Esta fuente de datos contiene capas paneuropeas de alta resolución que proporcionan información sobre las características específicas de la cubierta terrestre. Las capas de alta resolución se producen a partir de imágenes de satélite.

La capa de densidad de árboles utilizada es “HRLForestTCD2015”, cuyo título es “HRL Forest Tree Cover Density 2015” la cual a su vez consta de las

capas de estado que muestran el nivel de densidad de la cubierta de los árboles en un rango de 0-100%, en resolución espacial de 20m y proyección nacional para el año de referencia 2015 (+/- 1 año).

Esta Red Neuronal es especialmente interesante, ya que muestra cómo pueden adaptarse unos datos procesados originariamente por humanos a partir de datos satelitales, para que una inteligencia artificial extrapole ese conocimiento a datos de ortofotografía aérea con mayor resolución y en áreas nuevas.

La leyenda correspondiente a los datos utilizados para el entrenamiento y, por lo tanto, también a los mapas generados es la siguiente:

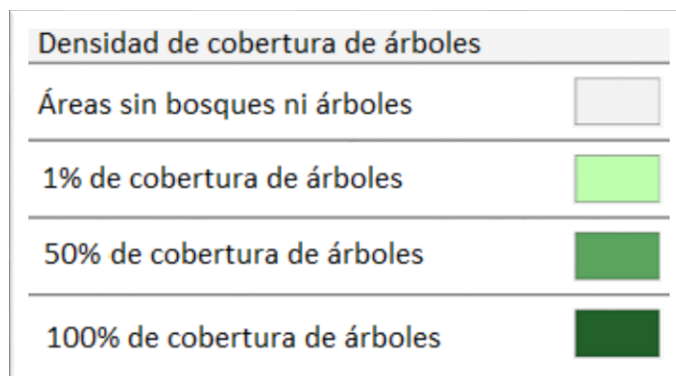


Figura 6.29. Leyenda del modelo DeepMapsBosques
Fuente: elaboración propia, a partir de los datos del IGN

Este modelo ha sido entrenado con un total de 6.480, obtenidas de forma aleatoria de todo el territorio nacional, con un nivel de zoom de 13 (escala aproximada 1:70.000) y habiéndose realizado el entrenamiento durante 2.200 iteraciones.

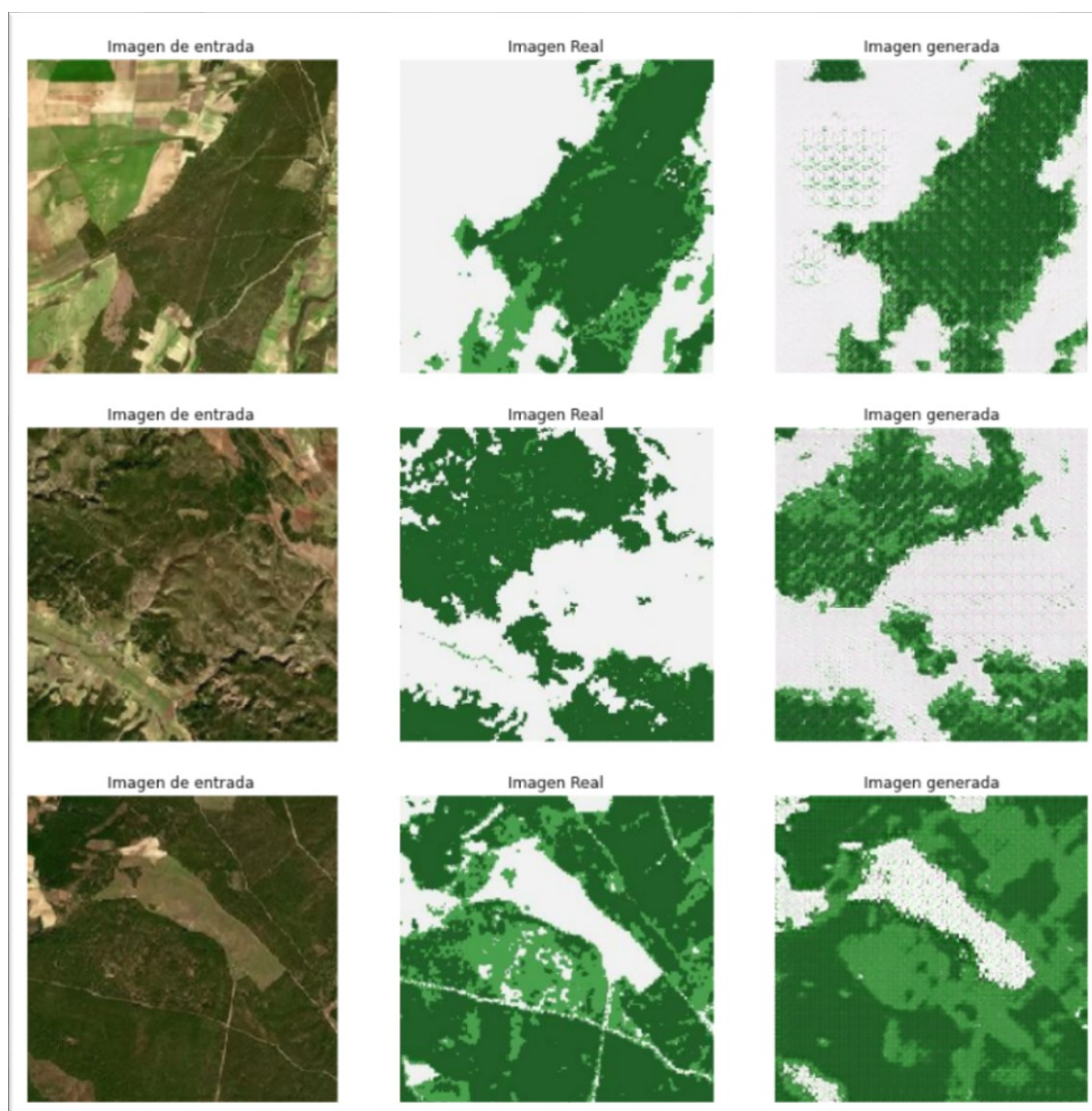


Figura 6.30. Resultados de la aplicación del modelo DeepMapsUsosNatura a Ortofotografías aéreas del PNOA
Fuente: elaboración propia, a partir de datos del IGN

Como puede observarse en los resultados obtenidos, la inteligencia artificial generada reconoce correctamente la cobertura de árboles, aportando incluso más precisión que la presente en los datos de entrenamiento, ajustando mejor no sólo las áreas cubiertas por árboles, sino también la densidad de los árboles en estas áreas, siendo capaz de aumentar la precisión de la representación mediante la creación de nuevas categorías mediante la combinación de las ya existentes. Gracias a las nuevas categorías creadas, la

inteligencia artificial puede expresar densidades de árboles que se encuentran entre los valores de las categorías originales. En este caso la autoemergencia de nuevos comportamientos en forma de la creación de categorías no se basa en la generación de colores intermedios, como en el caso de DeepMapsUsosNatura, sino en la creación de tramas que contienen los colores de las categorías existentes en los datos de entrenamiento, expresados en forma de patrones diferenciados, cada uno de ellos con mayor o menor porcentaje de los colores de las categorías originales en función de si la densidad de árboles real se aproxima más a una categoría o a otra.

6.4.3. DeepMapsUsosSIOSE

Este modelo ha sido entrenado con los datos de Información de Ocupación de Suelo de España (SIOSE) 2014 y de CORINE Land Cover 2018, obtenidos a través de uno de los WMS que ofrece como servicios el IDEE, de ocupación del suelo, formando los pares de imágenes necesarios para el entrenamiento con teselas extraídas del Servicio WMS del PNOA de Máxima Actualidad, a través del recurso [“https://www.ign.es/wms-inspire/pnoa-ma?SERVICE=WMS&”](https://www.ign.es/wms-inspire/pnoa-ma?SERVICE=WMS&), debido a que la escala a la que han sido extraídas las teselas es 1:73.000, las ortoimágenes corresponden al satélite Sentinel 2, ya que para escalas menores a 1:70.000, con el servicio de mapas web anteriormente citado se visualizan las imágenes de satélite Sentinel 2 de 10 metros de resolución y no las del ortoimágenes aéreas del PNOA.

El nombre de la capa de usos de los usos de suelo utilizada es “Usos de suelo existentes”, esta capa es servida mediante el servicio de visualización WMS 1.3.0 conforme al perfil Inspire de ISO 19128:2005. A este recurso puede

accederse a través de “<https://servicios.idee.es/wms-inspire/ocupacion-suelo?SERVICE=WMS&>”.

Tanto los datos de entrenamiento como los datos generados por la inteligencia artificial generada utilizan la nomenclatura de clases de Usos del Suelo HILUCS, distinguiendo las siguientes clases:

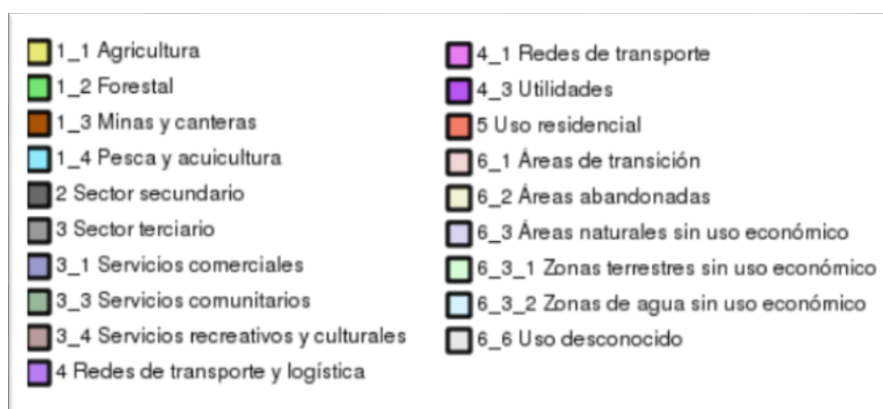


Figura 6.31. Legenda del modelo DeepMapsUsosSIOSE y de los datos con los que ha sido entrenado

Fuente: Elaboración propia, a partir de los datos del IGN

A continuación, se muestra un ejemplo de uno de los pares de imágenes utilizadas para el entrenamiento.

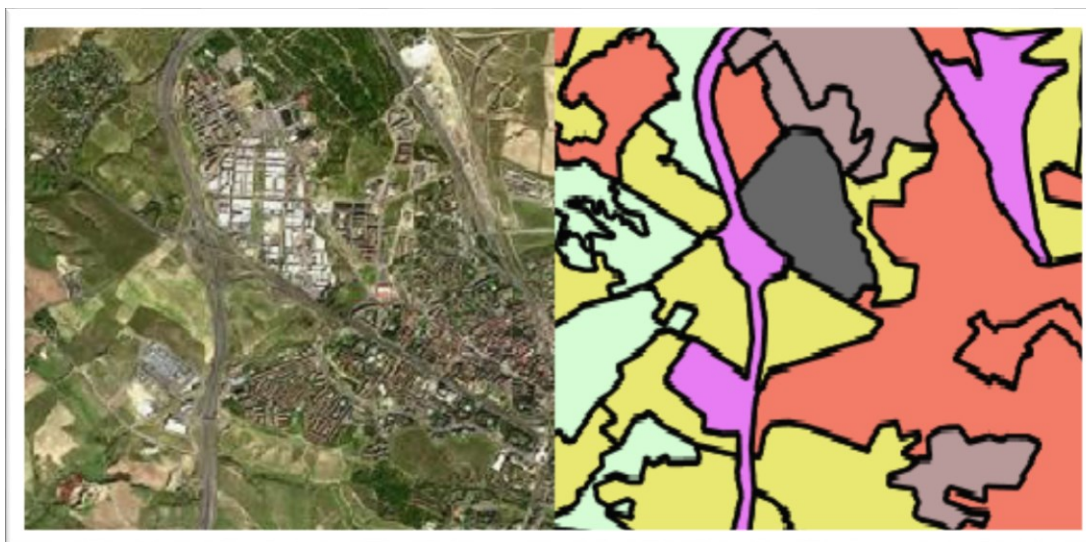


Figura 6.32. Ejemplo de par de imágenes utilizadas para el entrenamiento del modelo DeepMapsUsosSIOSE

Fuente: Elaboración propia a partir de los datos del IGN

El número de teselas con las que ha sido entrenado el modelo es de 1.860, obtenidas de forma aleatoria del territorio correspondiente al municipio de Madrid, con un nivel de zoom de 13 (escala aproximada 1:70.000) y el entrenamiento se realizó durante 1.250 iteraciones.

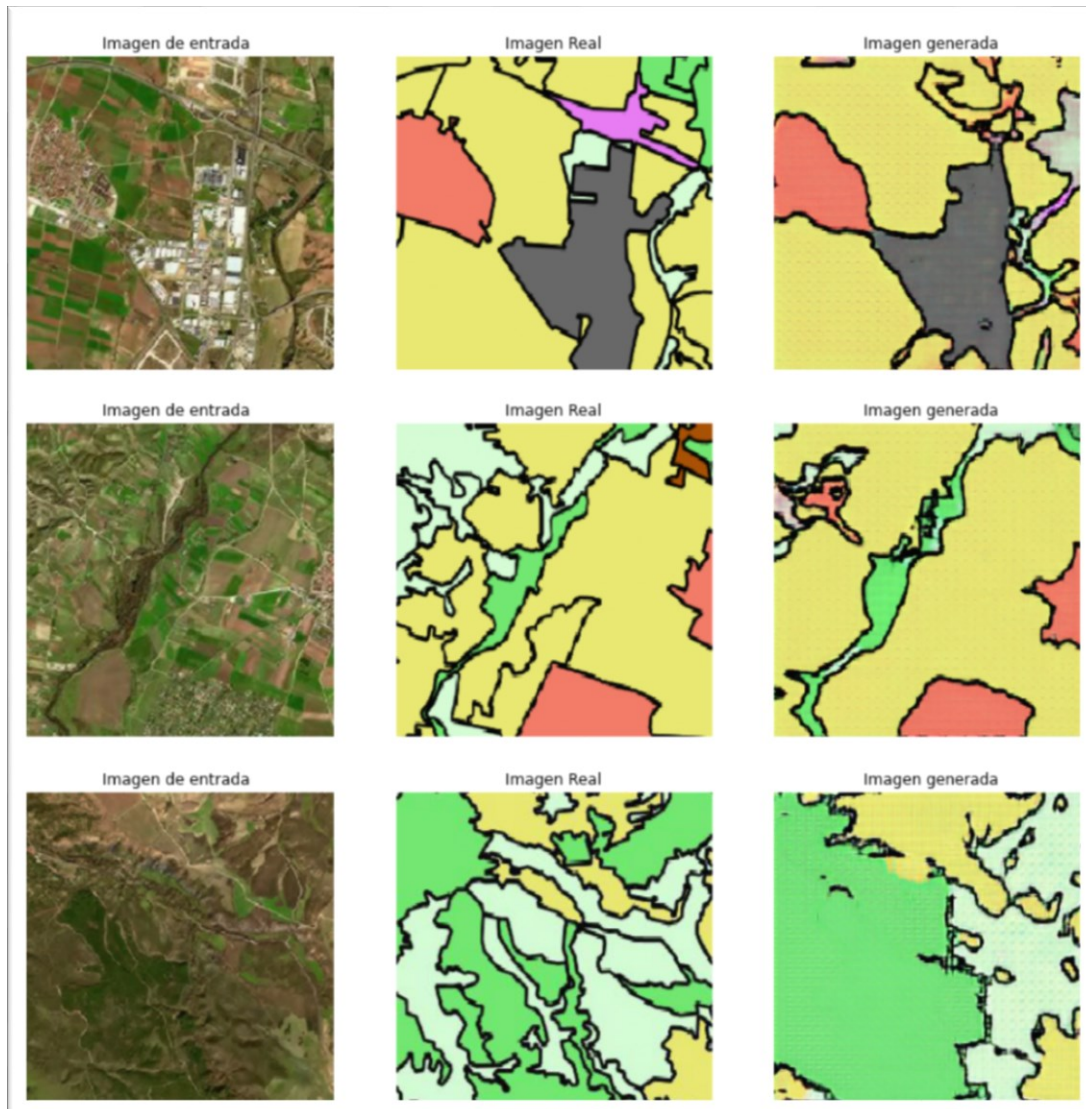


Figura 6.33. Resultados de la aplicación del modelo DeepMapsUsosSIOSE
Fuente: Elaboración propia, a partir de los datos del IGN

Los mapas generados por la Inteligencia Artificial DeepMapsUsosSIOSE tienen una escala menor a los generados por DeepMapsUsosNatura, por lo que muestran menor detalle que éstos. Sin embargo, siguen superando incluso la

elaboración de este tipo de mapas realizados por humanos, tal y como se muestra en el siguiente ejemplo en el cual la imagen generada por la inteligencia artificial DeepMapsUsosSIOSE ha sido capaz de crear una representación de la realidad más fiel que la desarrollada por humanos en la imagen original (Imagen Real).

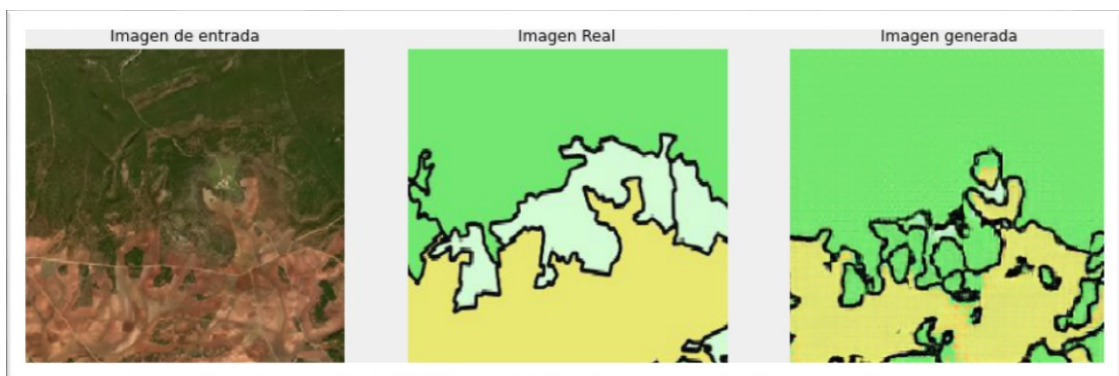


Figura 6.34. Comparativa entre los mapas de usos del suelo original del CORINE y el generado por DeepMapsUsosSIOSE, con respecto a una ortofotografía satelital real del terreno

Fuente: Elaboración propia, a partir de los datos del IGN

Como puede apreciarse en los ejemplos expuestos en este subepígrafe, la Inteligencia Artificial DeepMapsUsosSIOSE, genera mapas de usos del suelo que son una representación fiel de la realidad, a partir de ortoimágenes satelitales de Sentinel 2, con menor detalle que las ortoimágenes aéreas, en las cuales resulta difícil para los humanos distinguir con suficiente precisión los elementos presentes en las mismas para su posterior categorización. Las redes creadas, capaces de generar mapas temáticos, tienen una clara aplicabilidad, a pesar de haber sido desarrolladas principalmente para demostrar las posibilidades que ofrece el uso de estas tecnologías en la cartografía, siendo por lo tanto mejorables sus resultados si se dispusiera de más medios para su entrenamiento, ya que permitirían un gran ahorro de tiempo y recursos, debido a que pueden generar los mapas de forma automática, sin la necesidad de intervención humana, tanto a partir de ortoimágenes aéreas como satelitales.

6.4.4. DeepMapsSuperficies

El propósito de esta Inteligencia Artificial es la generación automática de mapas de superficies de cubierta terrestre, en base a los datos ocupación de suelo ofrecidos por IDEE, correspondientes a la capa “Ocupación de suelo de España”, la cual contiene información de Ocupación de Suelo de España (SIOSE) 2014 y CORINE Land Cover 2018. La denominación de las capas es conforme con las especificaciones de la Directiva Inspire 2007/2/EC.

Este modelo ha sido creado para trabajar con escalas comprendidas entre 1:500.000 y 1:70.000, el entrenamiento de este modelo se ha realizado partiendo de 1.076 pares de imágenes (Usos del suelo y Sentinel 2) durante 1.530 iteraciones.

Al igual que en la red DeepMapsUsosSIOSE, se utiliza la nomenclatura de clases de Usos del Suelo HILUCS, distinguiendo las siguientes clases:



Figura 6.35. Legenda del modelo DeepMapsSuperficies y de los datos con los que ha sido entrenado

Fuente: Elaboración propia, a partir de los datos del IGN y de IDEE

A los datos se ha accedido mediante el Servicio de Visualización Teselado, conforme al perfil INSPIRE de Web Map Tile Service (WMTS) 1.0.0. Las teselas

suministradas por este servicio tienen formato PNG y Sistema de Referencia por Coordenadas WGS84 Web Mercator (EPSG: 3857), por lo que se ha realizado una conversión con QGIS para ajustarlas al EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado), como las imágenes del satélite Sentinel 2 que se asociaron a éstas en el proceso de entrenamiento. Al servicio de mapas de ocupación del suelo se ha accedido mediante el siguiente WMTS “<http://servicios.idee.es/wmts/ocupacion-suelo?request=GetCapabilities&service=WMTS>”, creando una conexión con dicho servicio mediante QGIS, tal y como se muestra a continuación.

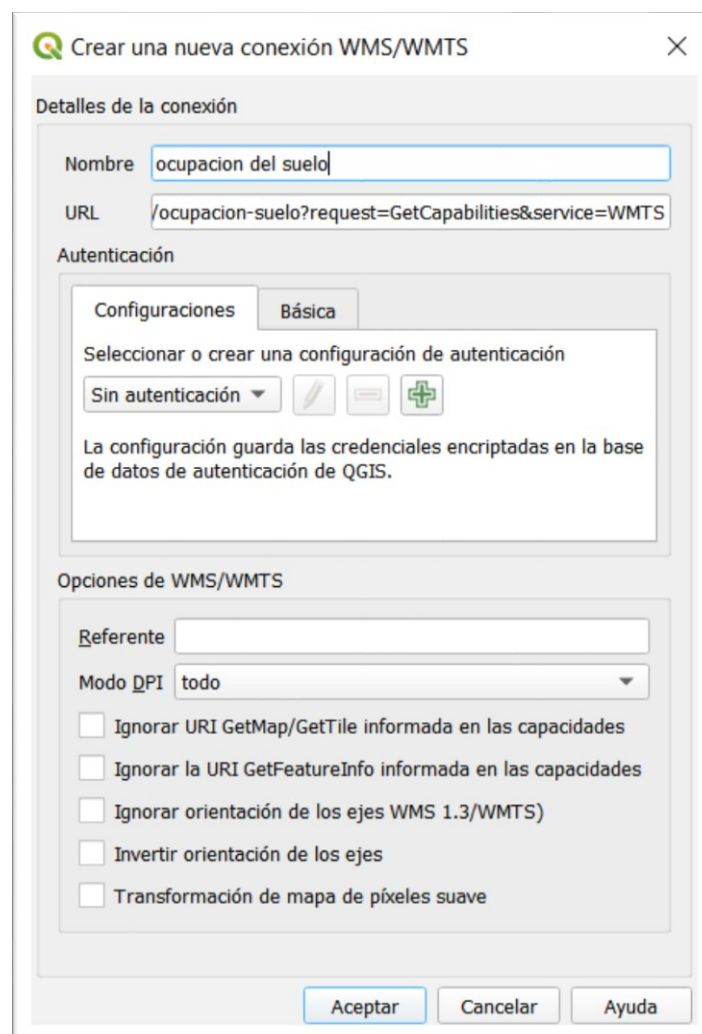


Figura 6.36. Introducción del servicio WMTS del IDEE de la capa de ocupación del suelo
Fuente: elaboración propia

El servicio WMS del cual se han obtenido los datos de Sentinel 2, es el que ofrece el IGN a través de “<https://www.ign.es/wms-inspire/pnoa-ma?SERVICE=WMS&>”, el cual suministra cobertura ráster opaca de imágenes de satélite y ortofotos PNOA de Máxima Actualidad (MA). Los rangos de visualización son: Imagen Sentinel 2 de 10 m de resolución hasta la escala aproximada 1:70.000; a partir de aquí ortofotografías PNOA MA de 0.25 m o 0.50 m de resolución, según la zona.

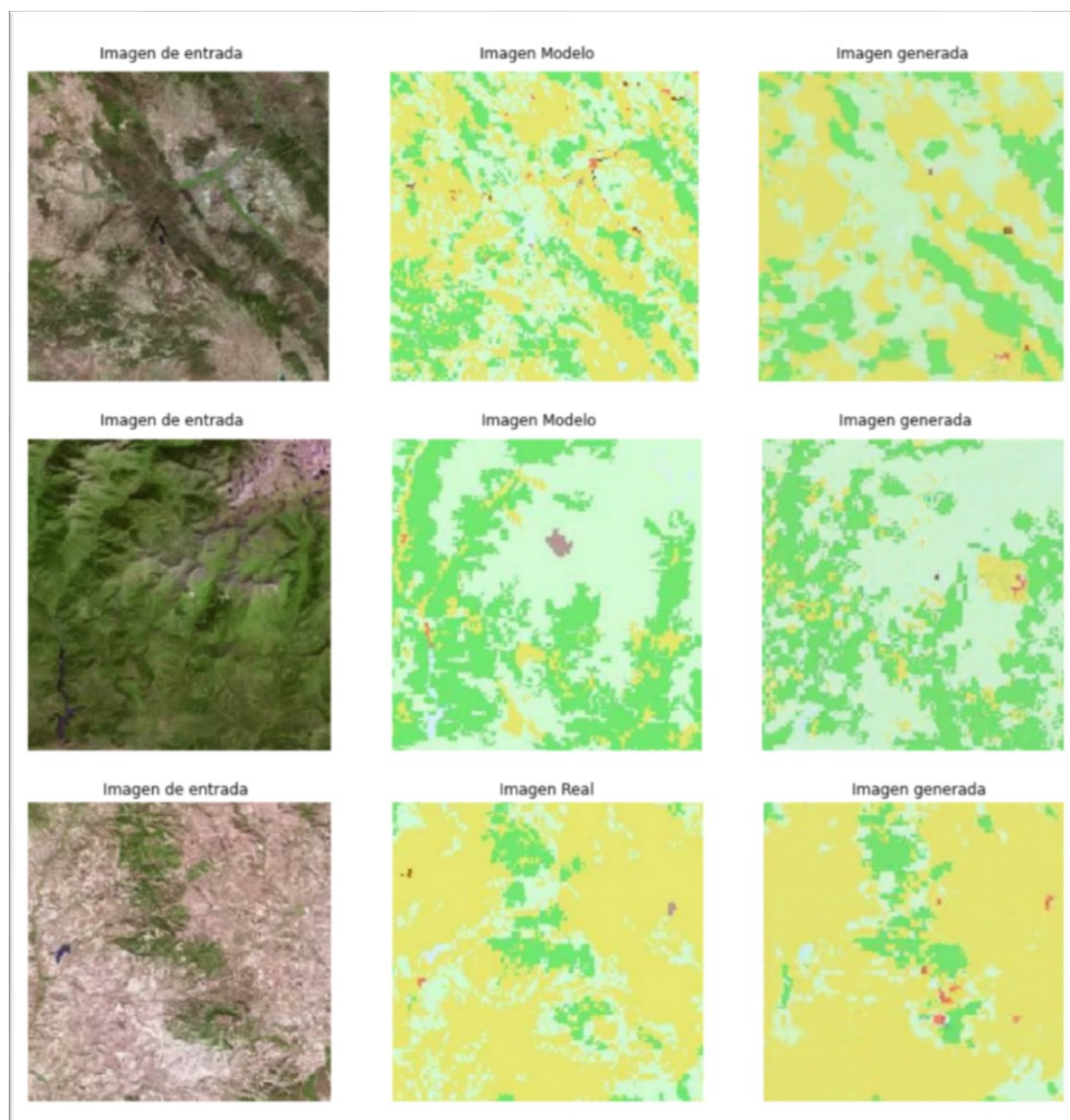


Figura 6.37. Resultados obtenidos con el modelo DeepMapsSuperficies
Fuente: elaboración propia, a partir de datos del IGN

7. Casos de uso de las inteligencias artificiales creadas

En este capítulo se exponen una serie de casos prácticos realizados utilizando la metodología propuesta en el capítulo 2; gracias a ello pueden comprobarse los beneficios potenciales de la aplicación de la Inteligencia Artificial en el mundo de la cartografía y la geografía. Este capítulo es especialmente relevante, ya que en él se muestran ejemplos de la aplicación de algunas de las redes desarrolladas para esta investigación a datos completamente diferentes a aquellos con los que han sido entrenadas.

A pesar de los éxitos obtenidos en las diferentes tareas que se muestran a continuación, debe destacarse que el propósito del desarrollo de inteligencias artificiales no es únicamente la creación de herramientas prácticas, sino principalmente establecer las bases para poder desarrollar toda una nueva generación de herramientas cartográficas.

Las distintas inteligencias artificiales que se han creado superan ampliamente los resultados obtenidos con otro tipo de técnicas, tanto automáticas como manuales, enfocadas a las mismas tareas, pero debido a las limitaciones de recursos con los que han sido creadas y entrenadas no muestran plenamente las posibilidades que ofrece la aplicación de redes generativas antagónicas en la cartografía.

Para que las inteligencias artificiales desarrolladas durante el presente trabajo desarrollasen todo su potencial, deberían ser entrenadas durante un número mayor de iteraciones con un número también mayor de datos de entrenamiento, para lo cual se hace imprescindible disponer de servidores con una mayor capacidad que a los que se ha tenido acceso.

Habiendo aclarado que los ejemplos incluidos en esta tesis y los distintos casos prácticos expuestos en este capítulo son sólo una pequeña muestra de las posibilidades de aplicaciones prácticas que abren el uso de la Inteligencia Artificial y el Big Data en cartografía, además de una demostración de su capacidad para el procesamiento de datos de fuentes y territorios diferentes a aquellos con los que fueron entrenadas, a continuación se exponen algunos de los resultados obtenidos en diferentes tareas en casos de uso reales.

En los casos prácticos que se muestran a continuación, se han seleccionado distintas áreas que ejemplifican bien las capacidades desarrolladas por las inteligencias artificiales creadas, las cuales van a ser procesadas por algunos de los modelos creados. Para la aplicación de los modelos, se ha seguido la metodología propuesta en el capítulo 2; sin embargo, dado que sólo se van a procesar determinadas áreas y no hojas enteras, el procedimiento seguido varía en algunos aspectos relativos al recorte de las capas ráster de origen. Es por ello por lo que en el primer caso se detallan estas particularidades del proceso aplicado a áreas concretas; este proceso de aplicación de las redes neuronales creadas a extensiones determinadas es el que se ha seguido para todos los casos prácticos expuestos en este capítulo.

7.1. Caso 1: Eliminación de parches en el mapa hipsométrico

En este caso se va a utilizar una de las inteligencias artificiales desarrolladas para eliminar errores y artefactos (DeepMapsReparaHipsométrico). Como ejemplo se han escogido datos del relieve terrestre obtenidos del SCUAM y generados a partir de datos ASTER del relieve del mundo.

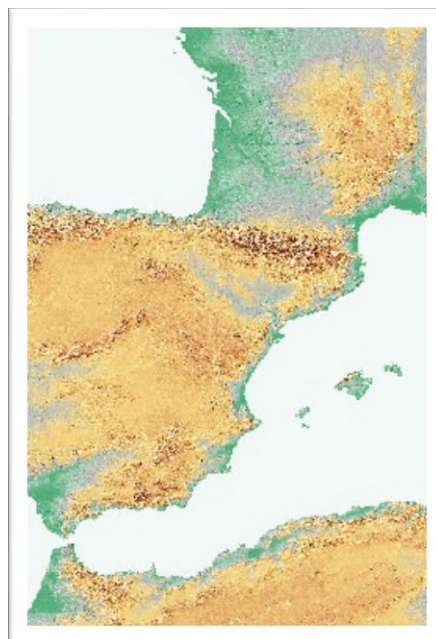


Figura 7.1: Contenido de la capa “a2_europa_occidental_relieveSCUAM_webmercator1” del SCUAM

Fuente: elaboración propia a partir de datos del IGN

En este caso, a diferencia de otros, el área escogida no pertenece a España, sino a Francia; de esta forma puede comprobarse cómo los modelos, una vez entrenados, son eficaces con los datos de lugares distantes a los lugares escogidos como entrenamiento.

Las imágenes proceden en concreto de la región comprendida en las coordenadas: 2240.770909090555, 180908.7709090906, 5790750.012296643, 5938471.688851668 [EPSG: 102100]



Figura 7.2: Localización del área a tratar
Fuente: elaboración propia a partir de datos del IGN

Para la limitación de los datos al área de interés que contiene los errores, antes de proceder a la generación de las teselas, como en los casos anteriormente expuestos, se ha procedido al corte de la capa original descargada del Servicio de Cartografía de la Universidad Autónoma de Madrid (a2_europa_occidental_relieveSCUAM_webmercator1).

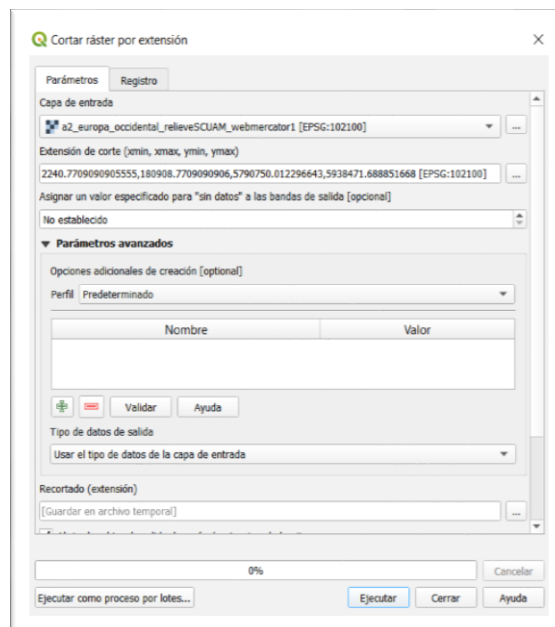


Figura 7.3: Corte de capa ráster
Fuente: elaboración propia

En esta área existen numerosos parches creados para ocultar artefactos presentes en los datos originales a partir de los cuales fue creado el mapa hipsométrico.

Como ya se vio en capítulos anteriores, los sistemas actuales para corregir este tipo de errores no dan resultados suficientemente verosímiles, a diferencia de lo que ocurre con las inteligencias artificiales desarrolladas en este trabajo. Como puede verse en muchas cartografías, existen numerosos parches que afean el aspecto de estas; estos parches son la forma tradicional de eliminar la presencia de artefactos, los cuales hasta ahora debían editarse manualmente. Gracias a las inteligencias artificiales desarrolladas, esta eliminación podría hacerse de forma automática y con un resultado mucho más verosímil que el aportado por un parche. Con varias de las redes neuronales aquí desarrolladas, incluso, pueden corregirse de forma automática las cartografías a las que se les han aplicado los parches, ofreciendo resultados que, si bien son recreaciones plausibles de la información faltante, resultan mucho más verosímiles y cercanas a la realidad que la ausencia de información en los píxeles, tal y como se muestra en este caso práctico y en los siguientes casos de este epígrafe.

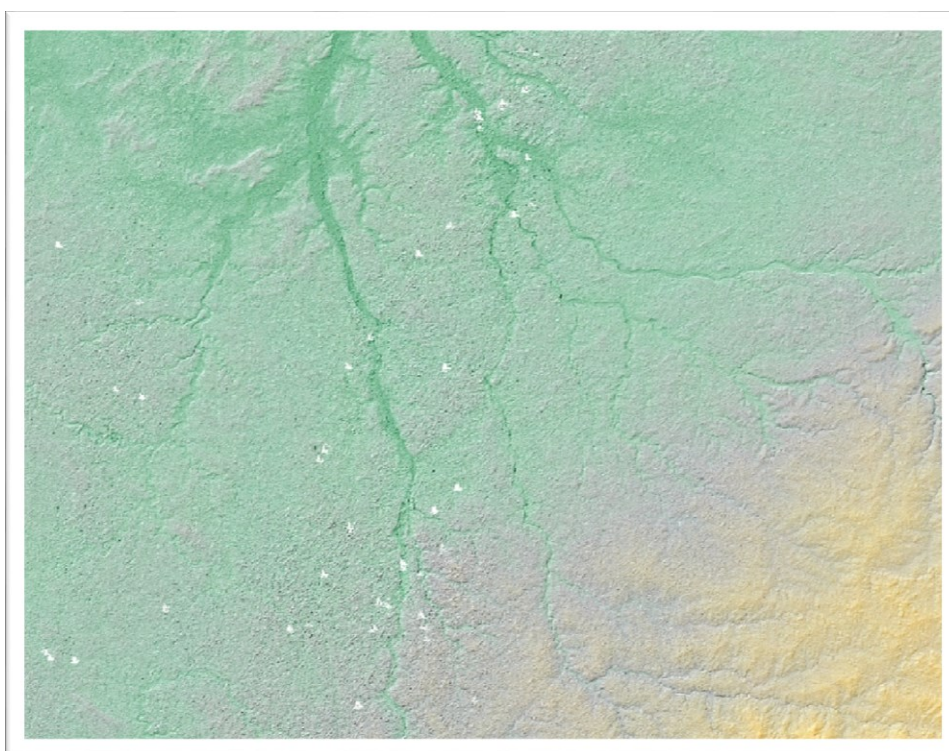


Figura 7.4: Área recortada para su procesamiento con el modelo
DeepMapsReparaHipsométrico
Fuente: elaboración propia a partir de datos del IGN

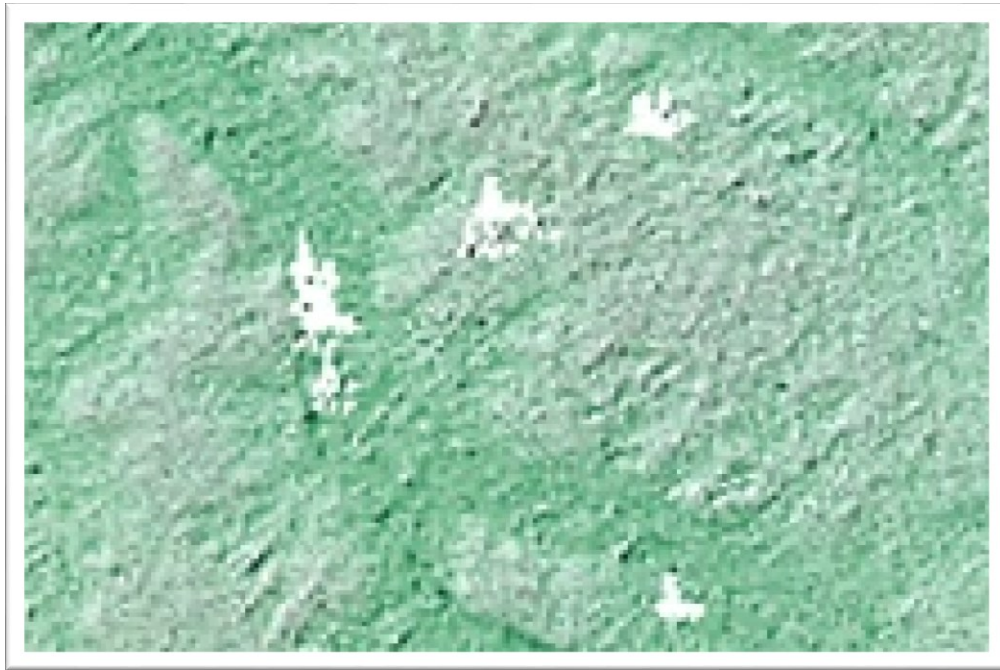


Figura 7.5: Detalle de algunos de los errores producidos por la presencia de parches
Fuente: elaboración propia a partir de datos del IGN

Una vez seleccionada el área a tratar, se procede a la generación de teselas, pero en esta ocasión se utiliza el sistema de coordenadas del archivo de origen, es decir Web Mercator, basado en WGS1984 en lugar de en ETRS89, concretamente se utiliza WGS_1984_Web_Mercator_Auxiliary_Sphere, cuyo identificador de sistema de referencia espacial es 102100. En este caso, el nivel de zoom con el que se van a generar las teselas es 13 (escala aproximada de 1:70.000), debido a que esta capa tiene una resolución menor a las tratadas en los casos anteriores.

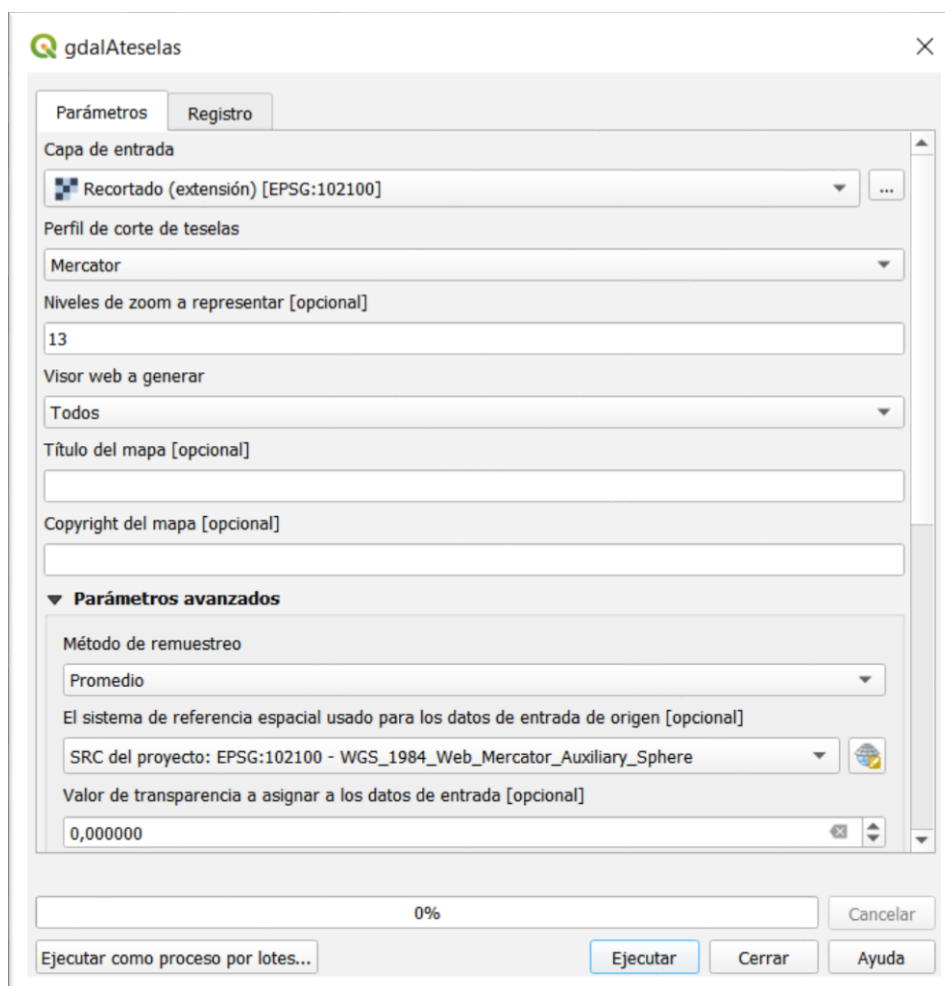


Figura 7.6: Generación de teselas
Fuente: elaboración propia

Elements SE (F:) > casosdeuso > europaoccidentalrelieveSCUAM > Caso Frances > 13 > input			
Nombre	Fecha de modificación	Tipo	Tamaño
.4096_5279.jpg	04/10/2020 17:16	Archivo JPG	3 KB
.4096_5279.kml	04/10/2020 17:16	KML	2 KB
.4096_5280.jpg	04/10/2020 17:16	Archivo JPG	6 KB
.4096_5280.kml	04/10/2020 17:16	KML	2 KB
.4096_5281.jpg	04/10/2020 17:16	Archivo JPG	6 KB
.4096_5281.kml	04/10/2020 17:16	KML	2 KB
.4096_5282.jpg	04/10/2020 17:16	Archivo JPG	7 KB
.4096_5282.kml	04/10/2020 17:16	KML	2 KB
.4096_5283.jpg	04/10/2020 17:16	Archivo JPG	6 KB
.4096_5283.kml	04/10/2020 17:16	KML	2 KB

Figura 7.7: Teselas y KML generados
Fuente: elaboración propia

Una vez extraídas las teselas y recopilados los archivos de imagen en un fichero .zip, se procede a su subida al servidor donde está alojado el modelo de corrección de errores (DeepMapsReparaHipsométrico).

Los archivos son procesados con el script de Python que se ha creado durante este trabajo para la aplicación de los modelos aquí desarrollados, tal y como se muestra en la siguiente imagen.

```
for example_input, example_target in test_dataset:
    generate_images(generator, example_input, example_target, nombres, numero
    )
    numero=numero+1

/content/drive/My Drive/nubesrelieve/. _4096_5279.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5280.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5281.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5282.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5283.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5284.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5285.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5286.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5287.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5288.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5289.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5290.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5291.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5292.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5293.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5294.jpg
/content/drive/My Drive/nubesrelieve/. _4096_5295.jpg
```

Figura 7.8: Procesado de teselas con inteligencia artificial
Fuente: elaboración propia

A continuación, se muestra el detalle de algunas de las correcciones realizadas en teselas individuales.

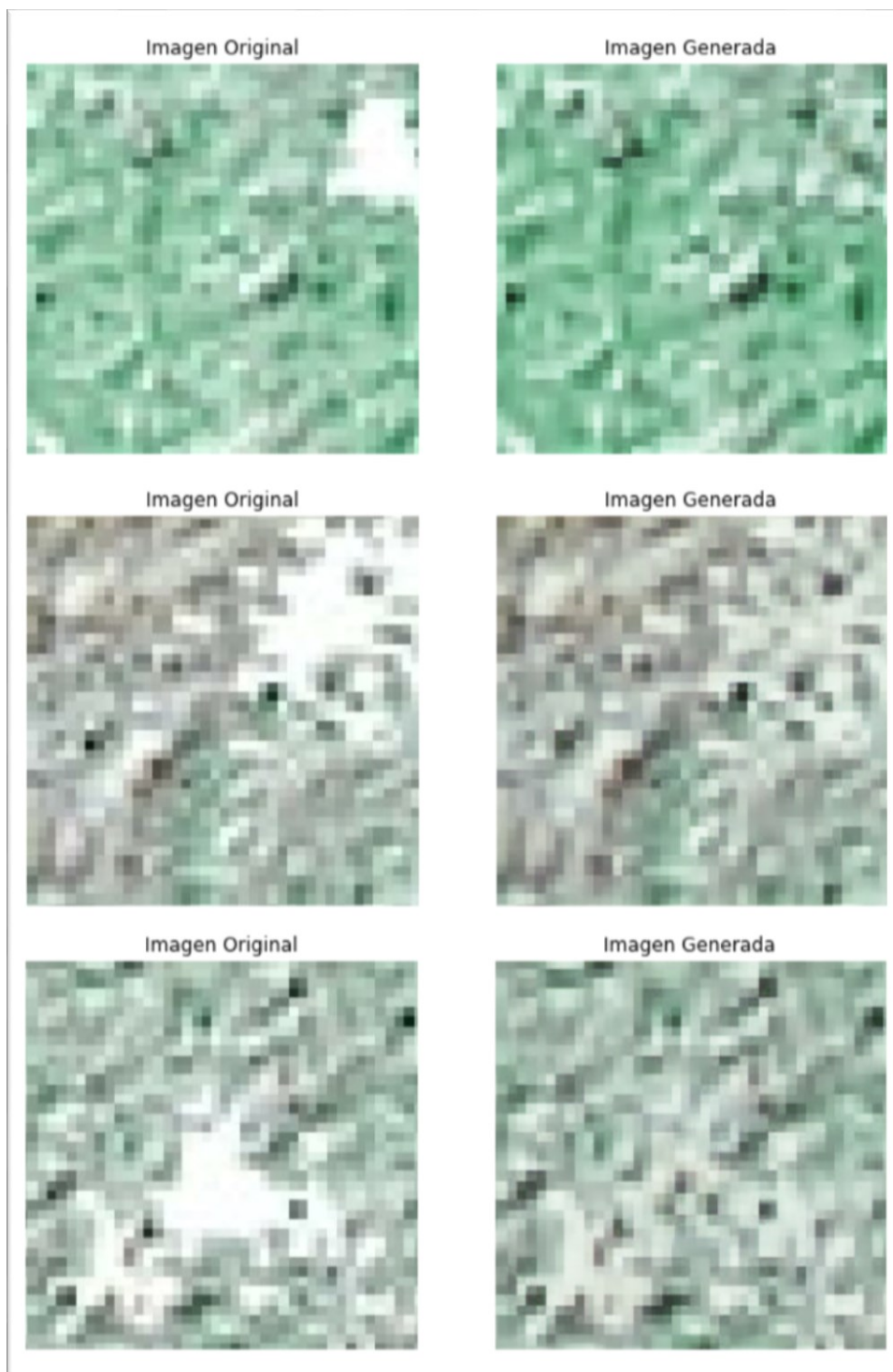


Figura 7.9: Ejemplos de teselas originales y generadas por el modelo
DeepMapsReparaHipsométrico
Fuente: elaboración propia a partir de datos del IGN

Una vez procesadas todas las teselas, se descargan del servidor en un archivo .zip, el cual se descomprime y se añaden a la misma carpeta los archivos .KML generados en pasos anteriores. Tras esto, desde QGIS, se procede a la combinación de todas las teselas georreferenciadas gracias a los archivos .KML.

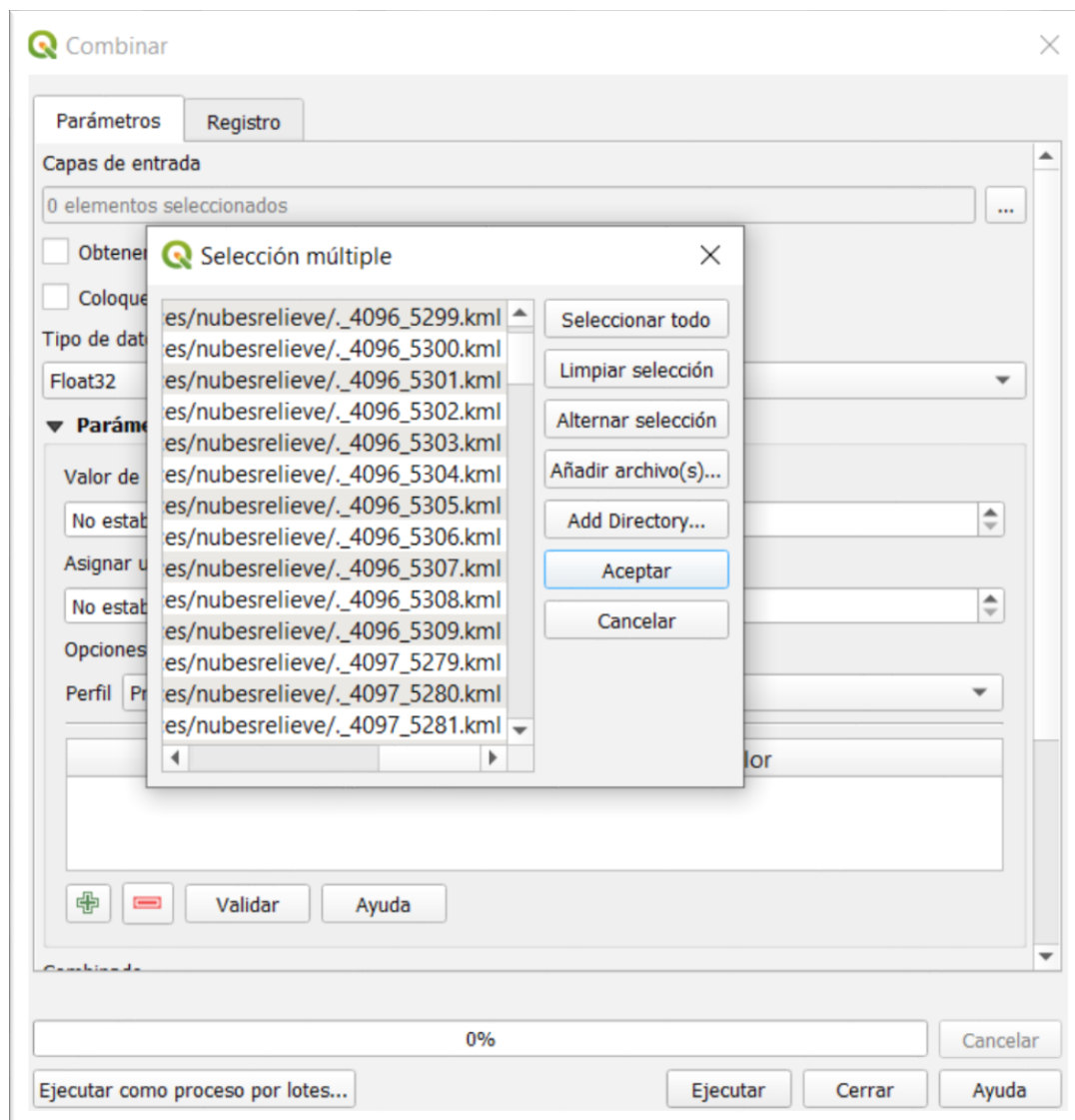


Figura 7.10: Combinación de teselas generadas por la inteligencia artificial
Fuente: elaboración propia

Una vez combinadas las teselas, quedan corregidos todos los errores existentes en el área seleccionada al comienzo de este subepígrafe.

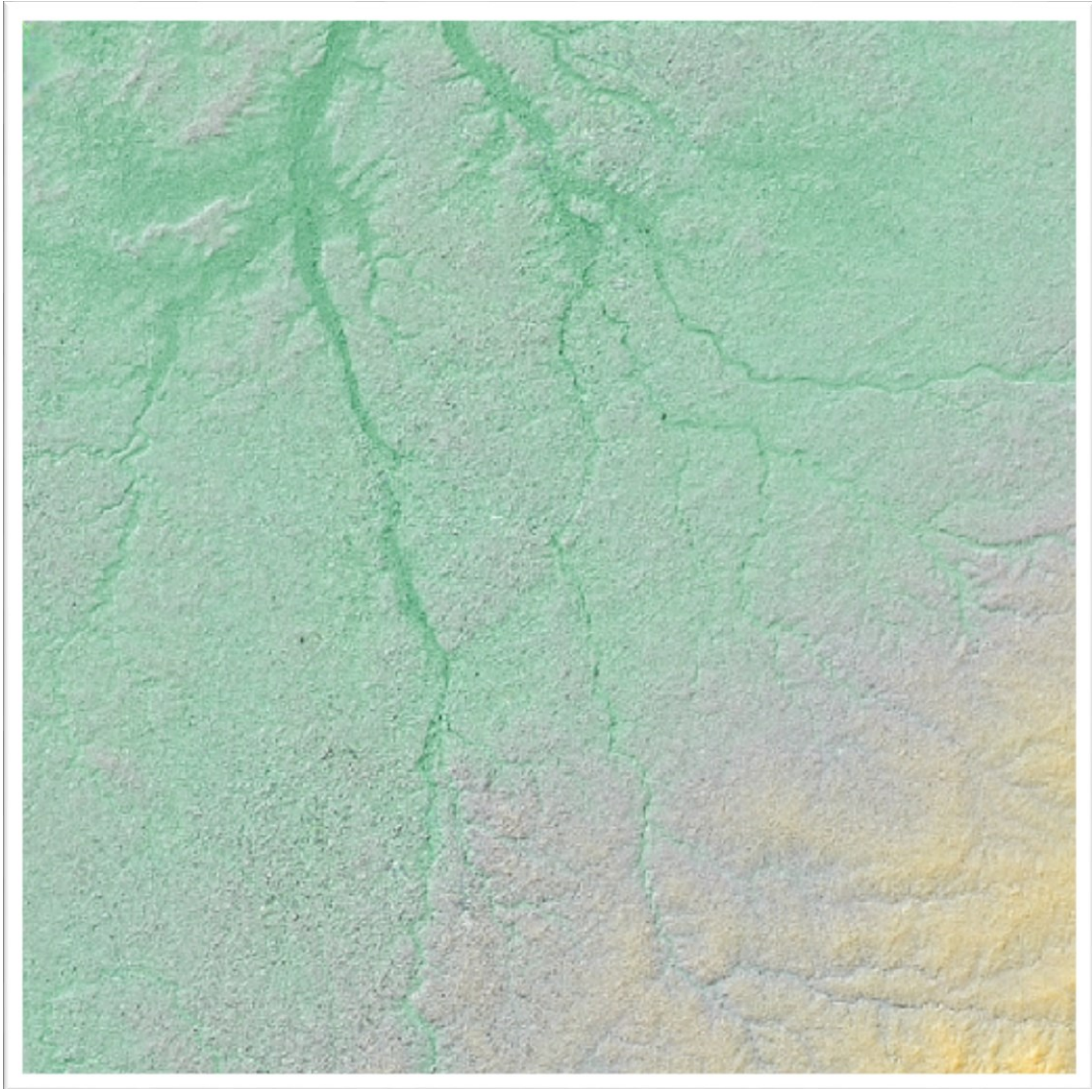


Figura 7.11. Unión de teselas corregidas
Fuente: elaboración propia a partir de datos del IGN

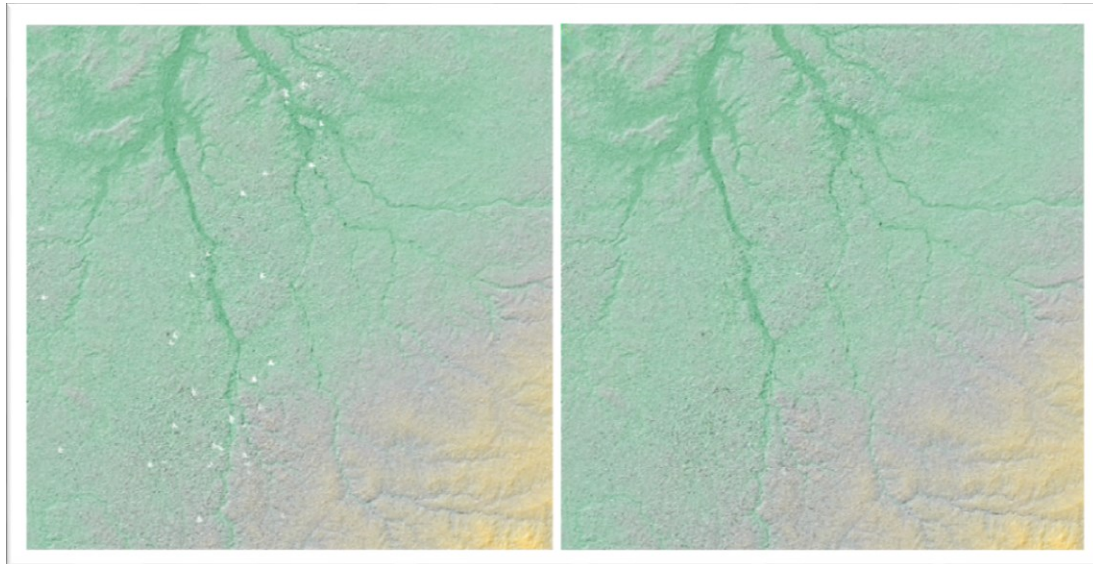


Figura 7.12. Comparativa entre la capa original y la corregida
Fuente: elaboración propia a partir de datos del IGN

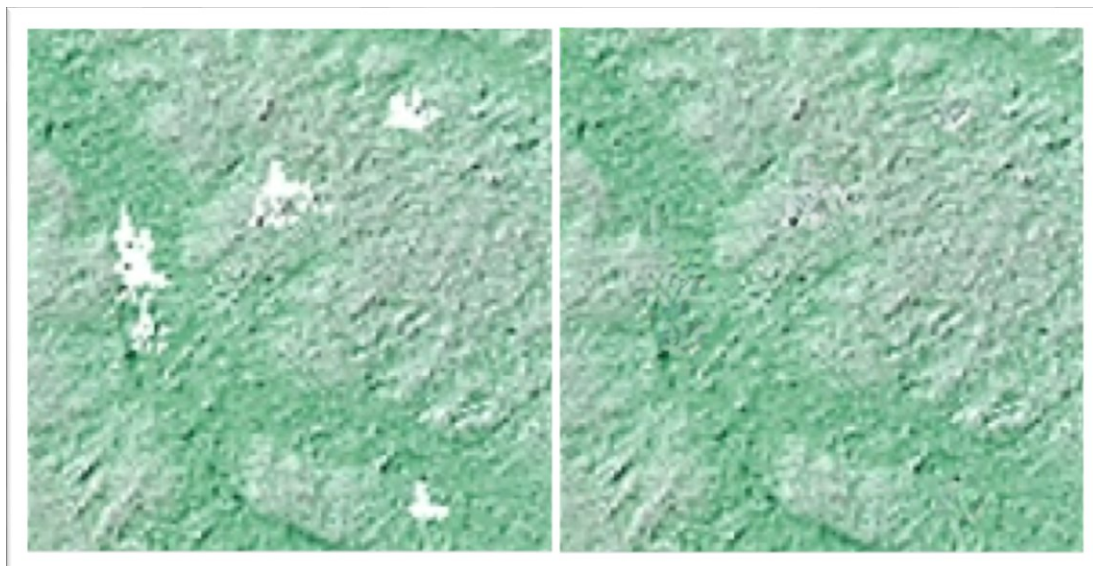


Figura 7.13. Detalle de comparativa entre imagen original e imagen corregida
Fuente: elaboración propia a partir de datos del IGN

7.2. Caso 2. Eliminación de parches de censura de datos

En los siguientes ejemplos se muestra la corrección de errores existentes derivados de la colocación de parches para censurar determinadas áreas, en las ortoimágenes del vuelo OLISTAT Oleícola y en el vuelo del PNOA del 2006. Los datos suministrados por el IGN en ambos casos tienen el SRC EPSG 25830 (ETRS89 / UTM zona 30N - Proyectado). Se han escogido estas dos fuentes de datos debido a que de esta forma puede ejemplificarse el desempeño de las inteligencias artificiales creadas durante esta investigación en datos de distintas características, ya que las ortoimágenes del OLISTAT son suministradas en escala de grises, mientras que las del PNOA2006 son ortoimágenes en RGB.

En los datos procedentes tanto del vuelo OLISTAT como del PNOA2006, existen varias zonas carentes de datos; para solucionar este defecto se ha aplicado la red neuronal DeepMapsInpainting, mediante la cual se ha podido generar un contenido plausible en aquellas zonas donde no había datos. A continuación, se exponen dos ejemplos de las correcciones realizadas en distintas áreas:

7.2.1. Caso 2. Ejemplo 1 - Recreación de datos correspondientes a las áreas censuradas entre Carabanchel y Pozuelo de Alarcón (Comunidad de Madrid)

Los datos censurados en este ejemplo corresponden al vuelo OLISTAT Oleícola, el cual fue realizado por el Ministerio de Agricultura entre 1997 y 1998 sobre parte del territorio español para contabilizar el número de olivos.

La siguiente ortoimagen corresponde al área de entre Carabanchel y Pozuelo de Alarcón (Comunidad de Madrid), comprendida en las coordenadas 428089.2416656523, 438312.7416760796, 4469125.345160385, 4477041.678501792 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)], con una escala de 1:50.000.

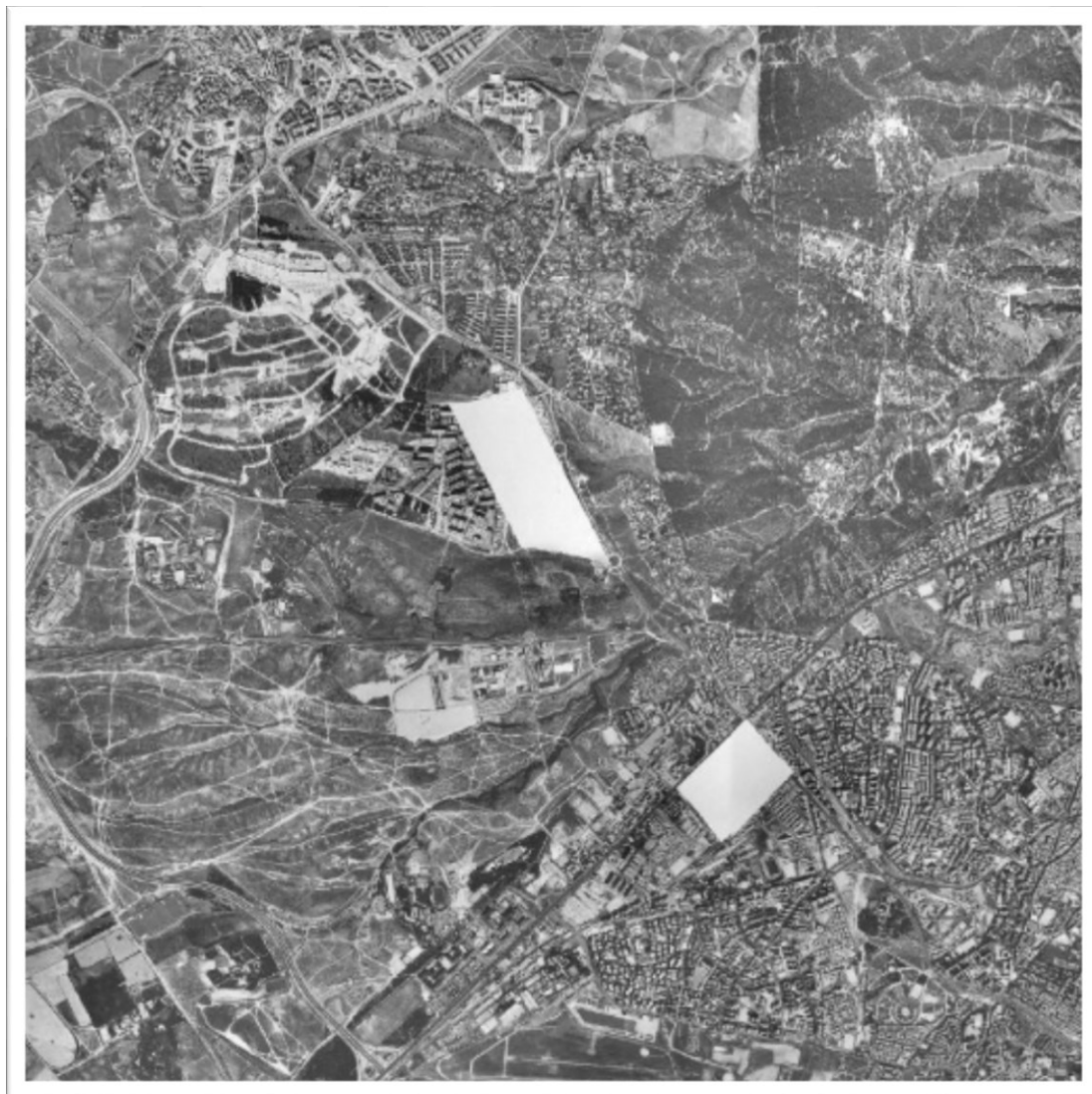


Figura 7.14. Ortoimagen original del OLISTAT en la que existen errores debidos a la ausencia de datos en determinadas áreas censuradas
Fuente: elaboración propia a partir de datos del IGN

La imagen anterior representa una fracción de la hoja MTN50 0559 del OLISTAT. En ella se aprecia claramente como existe una carencia de datos en las ubicaciones de las instalaciones de Radio Televisión Española de Prado del Rey, del Acuartelamiento “Capitán Sevillano” en Somosaguas y del Antiguo Acuartelamiento de Campamento.

Esta ausencia de datos probablemente responda a la aplicación de parches para censurar determinadas localizaciones por motivos de seguridad, sin embargo, la ausencia de datos en sí misma es un indicador de la existencia de información sensible ubicada en esos emplazamientos. Es por ello por lo que las inteligencias artificiales aquí propuestas pueden no sólo mejorar los datos faltantes sino también contribuir a la seguridad nacional haciendo inidentificables los lugares censurados, ya que los datos faltantes son completados con datos mucho más verosímiles que los parches o difuminaciones utilizados hasta ahora y por lo tanto mucho más difíciles de localizar e identificar. La eliminación de elementos de forma verosímil y prácticamente indistinguible para un humano ya se comentó y ejemplifiqué en el tercer epígrafe del capítulo 5, en el cual se muestra la eliminación de un edificio en concreto a una escala de 1:1.000, la cual permite apreciar mejor los detalles de los datos generados por la inteligencia artificial DeepMapsInpainting. A continuación, se muestra el resultado de la corrección realizada por la inteligencia artificial DeepMapsInpainting, aquí desarrollada.



Figura 7.15. Ortoimagen corregida mediante la aplicación de la inteligencia artificial
DeepMapsInpainting
Fuente: elaboración propia a partir de datos del IGN

En la imagen superior puede verse como las áreas carentes de datos han sido completadas con datos verosímiles, aportando un aspecto mucho mejor a la imagen y dificultando enormemente la labor de identificar aquellas áreas sensibles que habían sido originalmente censuradas.

7.2.2. Caso 2. Ejemplo 2: Recreación de datos correspondientes al área censurada de la Base de Apoyo de Radiotransmisores

Al igual que en el ejemplo anterior, los datos originales censurados, utilizados en este ejemplo, provienen del IGN y se corresponden con ortoimágenes del vuelo OLISTAT, pertenecientes a la hoja MTN50 0559.

En este ejemplo el error (o censura) está presente en los datos originales del OLISTAT al norte de Ciudad Universitaria, cerca del Real Club Puerta de Hierro, tal y como se muestra en la siguiente imagen.



Figura 7.16. Ortoimagen original de OLISTAT en la que existen errores debidos a la ausencia de datos en un área censurada

Fuente: elaboración propia a partir de datos del IGN

Las coordenadas de la ortoimagen original del OLISTAT (Figura 7.16), así como de la ortoimagen reparada generada por el modelo DeepMapsInpainting son las siguientes: 435740.9916734566, 439995.49167779676, 4477507.345168933, 4481296.178506131 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)]. Ambas ortoimágenes están representadas con una escala de 1:25.000.



Figura 7.17. Ortoimagen corregida mediante la aplicación de la inteligencia artificial
DeepMapsInpainting

Fuente: elaboración propia a partir de datos del IGN

En la imagen superior, puede comprobarse cómo la inteligencia artificial DeepMapsInpainting es capaz de completar de forma verosímil los datos ausentes debido a la censura, resultando muy difícil, cuando no imposible, la identificación del área censurada.

En otras ortofotografías más recientes, suministradas también por el IGN, se han intentado disimular algo más las censuras realizadas, sin embargo siguen resultando evidentes, ya que el método que ha sustituido la utilización de parches ha sido el difuminado de la imagen en el área a censurar y por lo tanto se aprecia una inconsistencia en el enfoque de la imagen que sigue siendo llamativa y fácilmente identificable, tal y como se muestra en la siguiente imagen, correspondiente a la Base de Apoyo de Radiotransmisores situada en la actual Calle de la Isla de Oza, vista anteriormente en este caso, pero con una escala de 1:6.000.



Figura 7.18. Comparativa de área censurada por PNOA de Máxima Actualidad (Izquierda) y la imagen generada por el modelo propio DeepMapsInpainting (Derecha)
Fuente: elaboración propia a partir de datos del IGN



Figura 7.19. Comparativa de área censurada por VirtualEarth de Microsoft (Izquierda) y la imagen generada por el modelo propio DeepMapsInpainting (Derecha)
Fuente: elaboración propia a partir de datos del IGN

Como puede comprobarse en los ejemplos anteriores, la aplicación de la inteligencia artificial DeepMapsInpainting mejora notablemente la verosimilitud en los métodos de censura más actuales.

7.2.3. Caso 2. Ejemplo 3: Recreación de datos correspondientes al área censurada de la Base de Helicópteros F.A.M.E.T

Este caso es un ejemplo de cómo en las ortoimágenes del PNOA también aparecen áreas censuradas con parches y de cómo la inteligencia artificial DeepMapsInpainting es capaz de rellenar los datos faltantes con datos verosímiles en ortoimágenes a color, tal como se mostraba en los datos de entrenamiento del capítulo 5.

En este caso, el área correspondiente a la Base de Helicópteros F.A.M.E.T., situada al norte de Colmenar Viejo (Madrid), ha sido censurada con un parche blanco. Los datos utilizados en este caso pertenecen al vuelo del PNOA2006, correspondientes a las siguientes coordenadas: 432905.48516015033, 439699.98516705947, 4500937.191546901, 4507064.941553132 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)]. La escala a la que se muestran las imágenes es de 1:50.000.



Figura 7.20. Ortoimagen original del PNOA2006 en la que existen errores debidos a la ausencia de datos en un área censurada

Fuente: elaboración propia a partir de datos del IGN

La imagen original (Figura 7.20), que contiene el parche, pertenece a la hoja MTN50 0534 del PNOA2006.

En la Figura 7.21 puede apreciarse como los datos ausentes han sido completados con datos plausibles, dando a la imagen censurada un aspecto mucho más realista y dificultando la identificación del área sensible.



Figura 7.21. Ortoimagen corregida mediante la aplicación de la inteligencia artificial
DeepMapsInpainting

Fuente: elaboración propia a partir de datos del IGN

7.2.3.1. Aplicaciones militares

Aunque la aplicación del modelo DeepMapsInpainting mejora mucho la verosimilitud de las ortoimágenes censuradas, durante las pruebas realizadas se han desarrollado otras inteligencias artificiales, a nivel experimental, en principio únicamente para profundizar en el conocimiento de las redes neuronales generativas antagónicas, las cuales podrían ser aplicadas para la censura en los mapas con resultados mucho más satisfactorios que cualquier otro método. Los resultados que aportan estas inteligencias artificiales superan cualquier otro método existente para la ocultación de infraestructuras, debido a que no se basan en rellenar un área con una información verosímil, ni por supuesto añadir un simple parche, sino que realizan una transformación de las estructuras de origen antrópico existentes a estructuras de origen natural de forma verosímil, de tal manera que las infraestructuras existentes en el área censurada, concuerdan con la estructura natural generada por la red.

Este concepto puede entenderse mejor a través de los siguientes ejemplos obtenidos mediante la aplicación de la red “DeepMapsOculta”, también creada durante esta investigación, sobre ortofotografías aéreas del PNOA con una escala de 1:1000, correspondientes a edificios e instalaciones deportivas del Instituto para la Formación Integral en Seguridad de la Comunidad de Madrid (441701.09953403426, 441892.65786756284, 4486650.254614285, 4486835.462947807 [EPSG: 25830]) y del IES Ciudad Escolar (442161.89786783734, 442353.4562013659, 4486195.1712804865, 4486380.379614009 [EPSG: 25830]), ambos situados en Valdelatas (Alcobendas, Comunidad de Madrid).



Figura 7.22. Ocultación de edificios e instalaciones deportivas del IFISE mediante la utilización de una inteligencia artificial de creación propia (DeepMapsOcultar)
Fuente: elaboración propia a partir de datos del IGN



Figura 7.23. Ocultación de edificios e instalaciones deportivas del IES Ciudad Escolar mediante la utilización de una inteligencia artificial de creación propia (DeepMapsOcultar)
Fuente: elaboración propia a partir de datos del IGN

Como puede apreciarse en los ejemplos anteriores los edificios e instalaciones deportivas han sido sustituidos por pastos y árboles, manteniendo la configuración general de las ortoimágenes, lo que permitiría una censura mucho más eficaz e indetectable, ya que las ortoimágenes censuradas mediante la aplicación de la inteligencia artificial DeepMapsOcultar serían coherentes con

los datos de menor resolución procedentes de otras fuentes, en los cuales no pueden apreciarse los detalles. Es decir, la inteligencia artificial DeepMapsOcultas, proporciona una interpretación plausible a los datos de menor resolución que los censurados, interpretación que no incluye edificios ni instalaciones importantes y en la que las carreteras han sido sustituidas por caminos.

Los detalles de esta inteligencia, al igual que otras muchas que han sido creadas durante esta tesis, con propósitos meramente experimentales, no se incluyen en el presente trabajo, debido a que se escapan al propósito principal de este; sin embargo, se han incluido los dos ejemplos anteriores de aplicación de DeepMapsOcultas para mostrar las posibilidades de aplicación militar de las inteligencias artificiales en la cartografía. En futuras publicaciones se expondrán los pormenores de ésta y otras muchas redes neuronales que han sido desarrolladas y cuya utilidad escapa a los propósitos de este estudio.

7.3. Caso 3: Eliminación de nubes en las imágenes del LandSat8

En este caso, el proceso de eliminación de imágenes es similar al expuesto en el caso anterior. Para este supuesto se ha elegido el mosaico nacional histórico del Landsat8 de 2014 descargado del IGN y cuyo nombre de archivo es

“pnt_landsat8_2014_mosaico_peninsula_illes_balears_b432_hu30_8bits_COG_90”. Estos datos han sido procesados mediante la red neuronal DeepMapsNubes descrita en el subepígrafe 5.3.5 del capítulo 5.

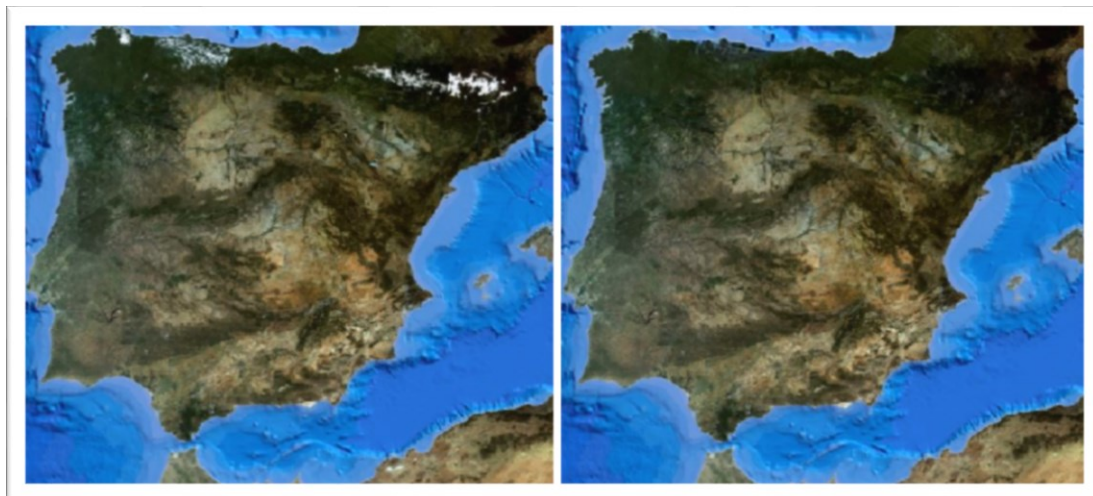


Figura 7.24. Comparativa entre la imagen original y la imagen tratada
Fuente: elaboración propia a partir de datos del IGN

A continuación, se muestra el detalle de algunas de las nubes eliminadas con una de las inteligencias artificiales generadas (DeepMapsNubes).



Figura 7.25. Ortofotografía satelital original del Landsat8 en la que se aprecia la presencia de
nubes

Fuente: elaboración propia a partir de datos del IGN



Figura 7.26. Imagen a la que se le ha aplicado el modelo de inteligencia artificial
DeepMapsNubes

Fuente: elaboración propia a partir de datos del IGN

Como puede apreciarse en los ejemplos mostrados anteriormente, la inteligencia artificial DeepMapsNubes realiza eficazmente la tarea de eliminación de nubes en las imágenes satelitales; sin embargo, elimina de la misma forma la nieve en los Pirineos, ya que la interpreta como si fuesen nubes que debe eliminar.

7.4. Caso 4: Corrección de defectos en las ortoimágenes del Vuelo Americano Serie B (1956-1957)

En los siguientes casos prácticos se muestra la corrección de errores existentes en las ortoimágenes del Vuelo Americano Serie B suministradas por el IGN, se trata de ortoimágenes del vuelo realizado por el Army Map Service de EEUU entre enero de 1956 y noviembre de 1957 sobre parte del territorio español, conocido como Vuelo Americano, serie B. A pesar de que en el momento de la captura de los datos en los años cincuenta, el SRC oficial era el EPSG 23030, los datos suministrados por el IGN tienen como SRC el EPSG 25830 (ETRS89 / UTM zona 30N - Proyectado).

7.4.1. Caso 4. Ejemplo 1: Corrección de anomalía cercana a la Plaza de Toros de las Ventas (Madrid)

La ortoimagen mostrada a continuación pertenece a la hoja 0559 del Vuelo Americano Serie B, en el área comprendida en las siguientes coordenadas 443343.790720078,445063.85100592003,4475251.376456745,4476282.6964 33822 [EPSG:25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.27. Ortofotografía original del Vuelo Americano Serie B con elemento extraño
Fuente: elaboración propia a partir de los datos del IGN

Tanto la imagen anterior como la imagen corregida generada están representadas a una escala de 1:6.000.

Como puede apreciarse en los distintos casos de uso analizados en este documento, los datos originales suministrados por el IGN tienen numerosos errores, algunos de ellos, como en este caso, se corresponden a elementos extraños presentes durante el proceso de digitalización de las ortofotografías originales. En este caso para corregir el error derivado de la presencia de lo que presumiblemente es un pelo en el proceso de escaneado de las fotografías, se ha aplicado la inteligencia artificial DeepMapsRuidoyDesenfoqueBN creada durante este trabajo, obteniendo el resultado que se muestra a continuación.



Figura 7.28. Ortofotografía del Vuelo Americano Serie B restaurada con el modelo
DeepMapsRuidoyDesenfoqueBN
Fuente: elaboración propia a partir de los datos del IGN

Tal y como se muestra en la imagen superior, el error ha sido corregido de forma automática por la inteligencia artificial DeepMapsRuidoyDesenfoqueBN; cabe mencionar que la capacidad de eliminar elementos ajenos a las ortoimágenes es una habilidad emergente que ha surgido de manera espontánea en muchas de las inteligencias artificiales desarrolladas en la presente investigación.

7.4.2. Caso 4. Ejemplo 2: Corrección de anomalía en el actual distrito de San Blas (Madrid)

En este caso, se muestra otro ejemplo del proceso de corrección de elementos extraños, en el cual también está presente un pelo en la ortoimagen. Este tipo de error, debido a la presencia de pelos, los cuales además presentan en todos los casos características similares es relativamente frecuente en las ortoimágenes pertenecientes al Vuelo Americano Serie B, lo cual parece indicar que este pelo estaba presente en el escáner durante el proceso de digitalización de las imágenes; esto ha provocado una merma importante en la calidad de los datos.

Una de las aplicaciones más interesantes que tienen las redes neuronales desarrolladas es la corrección automatizada de errores producidos por descuidos en el proceso de digitalización, sin tener que repetir la digitalización de todos los datos con el coste económico y de tiempo que ello supone.

En este caso el error se corresponde con la actual zona del Parque del Paraíso, en San Blas (Madrid) y, como en el caso anterior, la presencia de un pelo estropea la ortoimagen. Este defecto ha podido ser corregido mediante la aplicación de la red neuronal DeepMapsRuidoyDesenfoqueBN.

Las coordenadas de las ortoimágenes, tanto original del Vuelo Americano Serie B, como la corregida generada por la inteligencia artificial DeepMapsRuidoyDesenfoqueBN, son: 446623.96120272466, 447753.1610850332, 4475357.611963643, 4476255.242314061 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)], y la escala de estas es de 1:6.000. Los datos originales han sido extraídos de la hoja MTN50 0559 del Vuelo Americano Serie B.



Figura 7.29. Ortofotografía original del Vuelo Americano Serie B con elemento extraño
Fuente: elaboración propia a partir de los datos del IGN



Figura 7.30. Ortofotografía del Vuelo Americano Serie B restaurada con el modelo
DeepMapsRuidoyDesenfoqueBN

7.4.3. Caso 4. Ejemplo 3: Corrección de roturas en ortofotografía de las áreas de Manoteras y Pinar de Chamartín (Madrid)

En este caso, a diferencia de los anteriores, se ha procedido a la eliminación de los defectos derivados de roturas presentes en el soporte físico original. Con este ejemplo puede comprobarse cómo las inteligencias artificiales desarrolladas no sólo son capaces de corregir defectos en el proceso de digitalización, sino de recuperar elementos ya deteriorados en origen.

Las ortoimágenes original y corregida por la inteligencia artificial representan el área comprendida en las siguientes coordenadas: 442377.22665590607, 444449.4158691028, 4479870.979727899, 4481460.931359223 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)], con una escala de 1:10.000. Los datos originales han sido extraídos de la hoja MTN50 0559 del Vuelo Americano Serie B.

Como puede verse en la siguiente ortoimagen, en ocasiones las ortofotografías antiguas presentan daños físicos como roturas y arañazos, un problema que hasta ahora no ha sido resuelto; sin embargo, mediante la aplicación de la inteligencia artificial DeepMapsRuidoyDesenfoqueBN, se ha podido recuperar la imagen sin deterioros físicos.



Figura 7.31. Ortofotografía original del Vuelo Americano Serie B con roturas
Fuente: elaboración propia a partir de los datos del IGN



Figura 7.32. Ortofotografía del Vuelo Americano Serie B restaurada con el modelo
DeepMapsRuidoyDesenfoqueBN
Fuente: elaboración propia a partir de los datos del IGN

La Figura 7.32 muestra cómo se han corregido de forma automática los daños físicos presentes en los datos originales del IGN (Figura 7.31).

7.4.4. Caso 4. Ejemplo 4: Corrección de arañazo en ortofotografía de área situada entre Fuenlabrada y Leganés (Madrid)

Como ya se ha visto en los casos anteriores, las ortoimágenes del Vuelo Americano Serie B presentan numerosos daños, en parte debidos a su mala conservación; en este caso se aplica la red neuronal DeepMapsRuidoyDesenfoqueBN para corregir un arañazo presente en el área que actualmente ocupa el Parque de Polvoranca, situado al suroeste de Leganés (Madrid), concretamente en las coordenadas 432370.20000335167, 432715.76381511457, 4463644.401292299, 4464018.016052512 [EPSG:25830 (ETRS89 / UTM zona 30N - Proyectado)]. Los datos originales han sido extraídos de la hoja MTN50 0582 del Vuelo Americano Serie B.

Tanto la ortofotografía original, como la restaurada están representadas con una escala de 1:25.000.

La siguiente imagen muestra el defecto debido al arañazo, anteriormente comentado, el cual atraviesa distintos campos de labor, por lo que a la inteligencia artificial le resulta sencillo identificarlo como una anomalía y corregirlo, tal y como aparece en la siguiente imagen.



Figura 7.33. Ortofotografía original del Vuelo Americano Serie B con arañazo
Fuente: elaboración propia a partir de los datos del IGN



Figura 7.34. Ortofotografía del Vuelo Americano Serie B restaurada con el modelo
DeepMapsRuidoyDesenfoueBN
Fuente: elaboración propia a partir de los datos del IGN

A lo largo de esta investigación se ha podido comprobar cómo los errores mostrados en los casos anteriores correspondientes a hojas MTN50, correspondientes al vuelo americano, descargadas a través del centro de descargas del IGN, también están presentes en los datos suministrados a través de los servicios WMS, del IDEE y del IGN.

En el caso del servicio WMS proporcionado por el IGN por el servicio disponible a través de <http://www.ign.es/wms/pnoa-historico>. Este servicio proporciona imágenes históricas, ortorectificadas, de distintos vuelos fotográficos de cobertura nacional promovidos por las Administraciones Públicas. Se muestran las ortofotos de los siguientes vuelos: PNOA (una capa por cada año de vuelo), Vuelo SIGPAC, Vuelo OLISTAT, Vuelo Americano Serie B, Vuelo Nacional y las del Vuelo Interministerial. En los casos mostrados anteriormente se han utilizado las imágenes correspondientes al Vuelo Americano Serie B de 1956-1957.

7.5. Caso 5: Corrección defectos en las ortoimágenes del vuelo de Ruiz de Alda (1929-1930)

Como puede comprobarse con una rápida revisión de los datos disponibles del vuelo de Ruiz de Alda, es habitual la presencia de polvo y desperfectos; debido a ello, puede resultar de gran utilidad una herramienta que permita su restauración de forma automatizada. Como puede comprobarse en este caso y en otros expuestos en este capítulo, las inteligencias artificiales desarrolladas son capaces de realizar esa tarea.

Los datos originales con los defectos que se han corregido en los ejemplos de este caso han sido extraídos de las ortofotografías aéreas del Vuelo Ruiz de Alda a las que se ha accedido mediante el servicio WMTS de la Confederación Hidrográfica de la Cuenca del Segura (<https://www.chsegura.es/wmts/ralda29?>), el sistema de referencia de coordenadas de las ortoimágenes que ofrece este servicio es el EPSG:25830 (ETRS89 / UTM zona 30N - Proyectado).

7.5.1. Caso 4. Ejemplo 1: Eliminación de polvo en ortofotografía de área situada al norte de San Vicente del Raspeig (Alicante)

En este ejemplo se ha aplicado el modelo DeepMapsRuidoyDesenfoqueBN a la ortofotografía del vuelo de Ruiz de Alda correspondiente a unos campos de labor ubicados al norte de San Vicente del Raspeig (Alicante), en la cual la presencia de polvo durante el proceso de escaneado de la misma ha provocado que los datos disponibles digitalmente tengan un aspecto descuidado.

El área tratada en este ejemplo está limitada por las siguientes coordenadas: 715724.3199063789, 715904.336476187, 4255032.118567358, 4255203.22342579 [EPSG: 25830].

La escala de la ortofotografía original, así como de la ortofotografía restaurada es de 1:1.000.

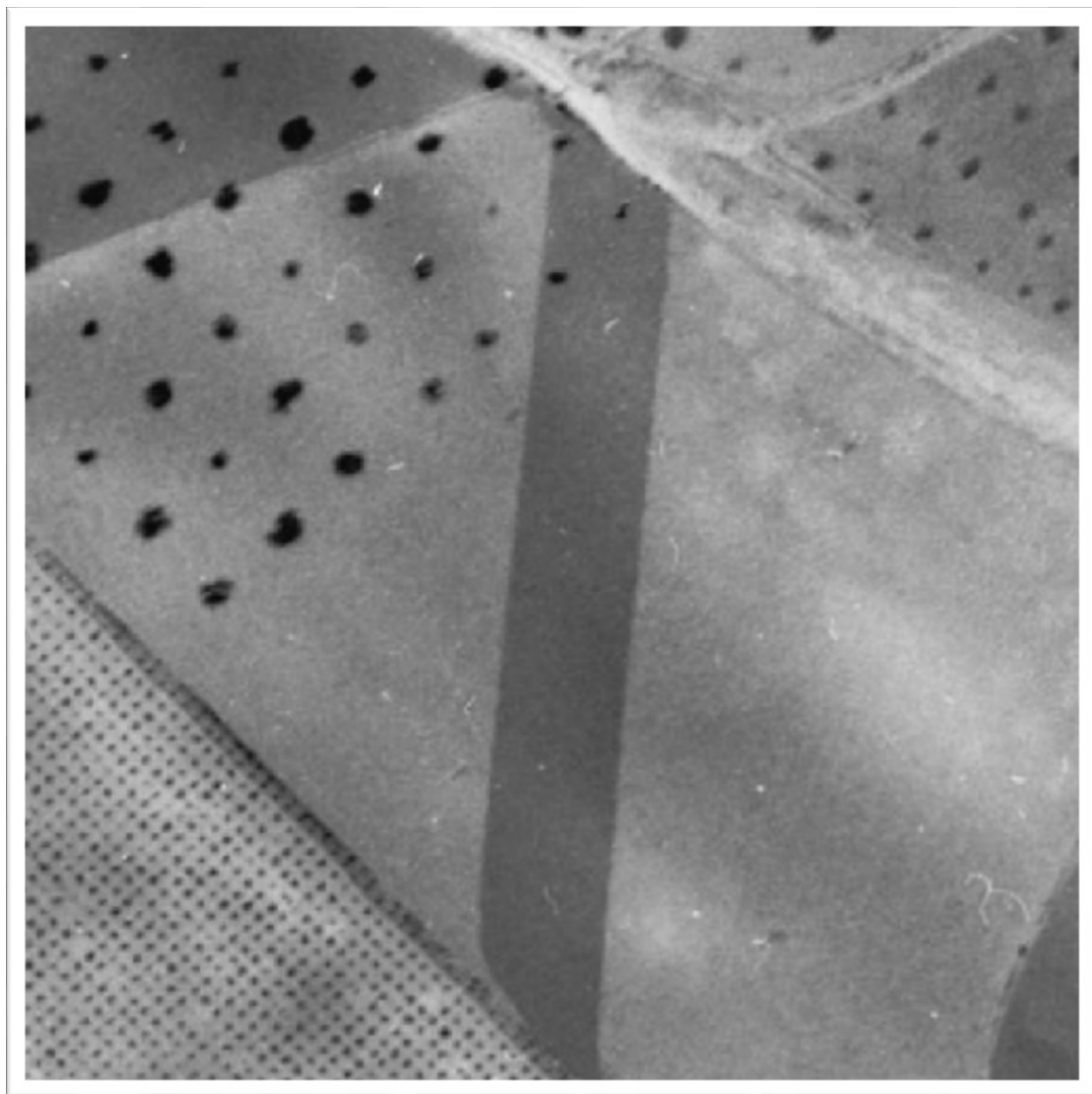


Figura 7.35. Ortofotografía original del Vuelo de Ruiz de Alda con polvo
Fuente: elaboración propia a partir de los datos de la Confederación Hidrográfica de la Cuenca del Segura

Una vez procesada la imagen mediante la red neuronal DeepMapsRuidoyDesenfoqueBN, puede comprobarse cómo ha mejorado su calidad y han sido eliminadas las imperfecciones existentes en la misma, tal y como puede apreciarse en la siguiente imagen.

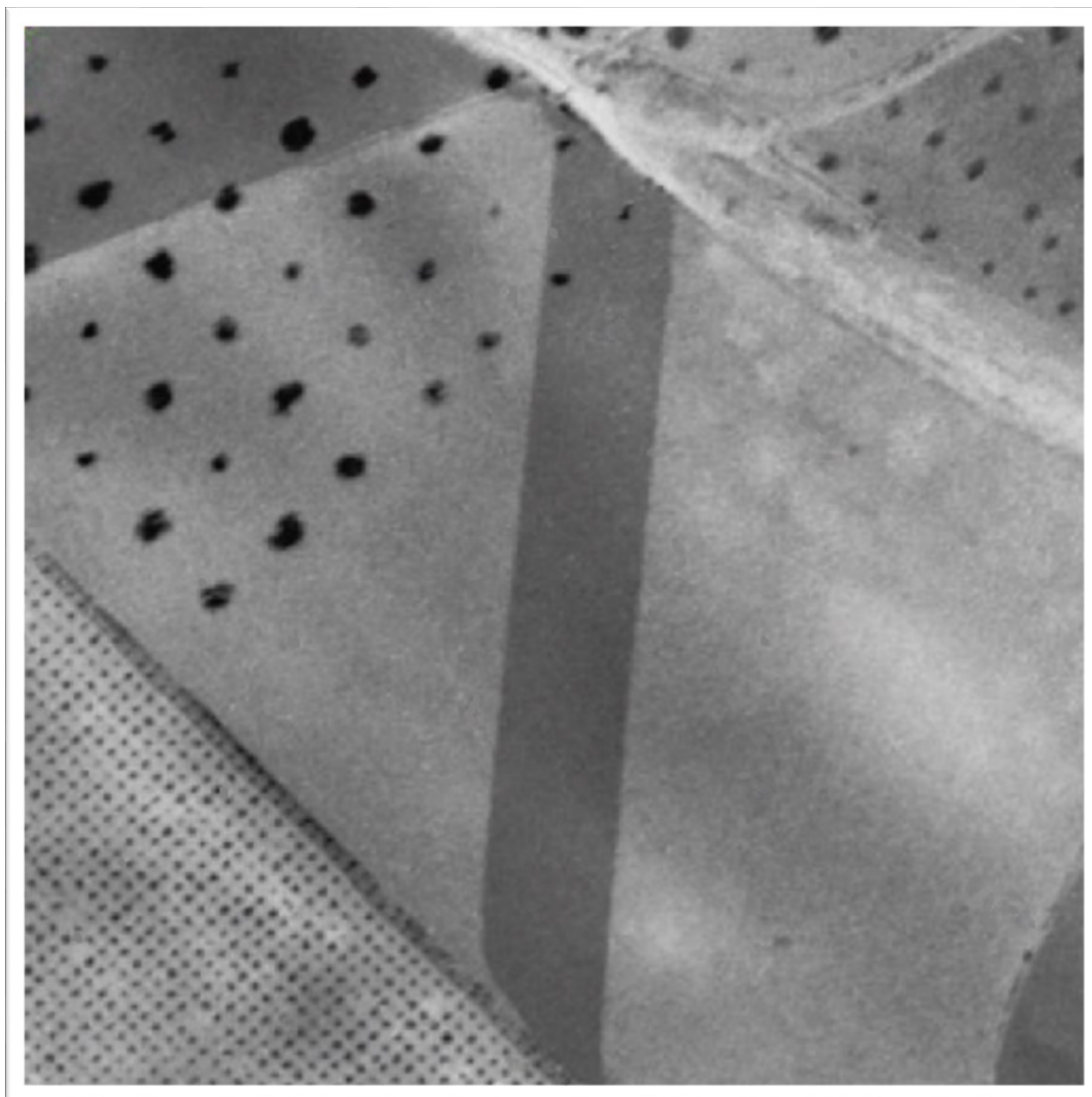


Figura 7.36. Ortofotografía del Vuelo de Ruiz de Alda restaurada con el modelo
DeepMapsRuidoyDesenfoqueBN
Fuente: elaboración propia a partir de los datos de la Confederación Hidrográfica de la Cuenca
del Segura

7.5.2. Caso 5. Ejemplo 2: Eliminación de polvo y arañazos en ortofotografía de Elche

En este caso los deterioros sufridos por la ortofotografía del Vuelo de Ruiz de Alda son aún mayores que en la anterior, ya que a la presencia de polvo en la imagen se le añaden otros desperfectos como los arañazos.

La zona corregida en este caso corresponde a campos de labor situados cerca de la actual Ciudad Universitaria de Elche (Alicante), concretamente a la zona situada en las siguientes coordenadas: 702594.3240253497, 702771.4890255304, 4239828.885017488, 4239993.477017656 [EPSG:25830 (ETRS89 / UTM zona 30N - Proyectado)].

Las imágenes mostradas a continuación, tanto la ortofotografía original como la que ha sido reparada mediante la aplicación del modelo DeepMapsRuidoyDesenfoqueBN, tienen una escala de 1:1.000.

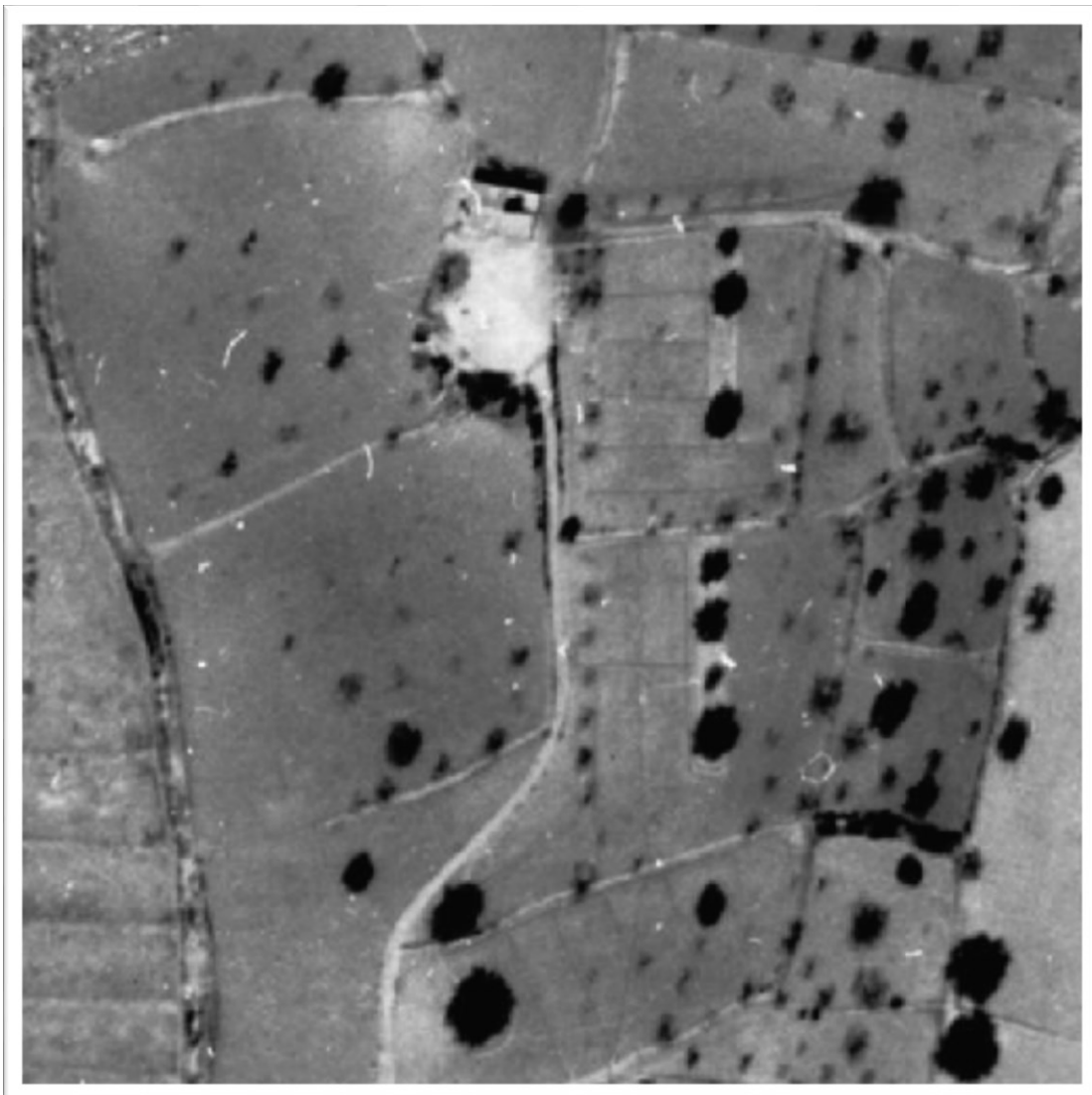


Figura 7.37. Ortofotografía original del Vuelo de Ruiz de Alda con polvo y arañazos
Fuente: elaboración propia a partir de los datos de la Confederación Hidrográfica de la Cuenca del Segura

A pesar de los graves desperfectos que muestra la ortofotografía original mostrada anteriormente, el modelo DeepMapsRuidoyDesenfoqueBN, es capaz de corregir todos ellos de forma automatizada, tal y como se muestra en la siguiente imagen.

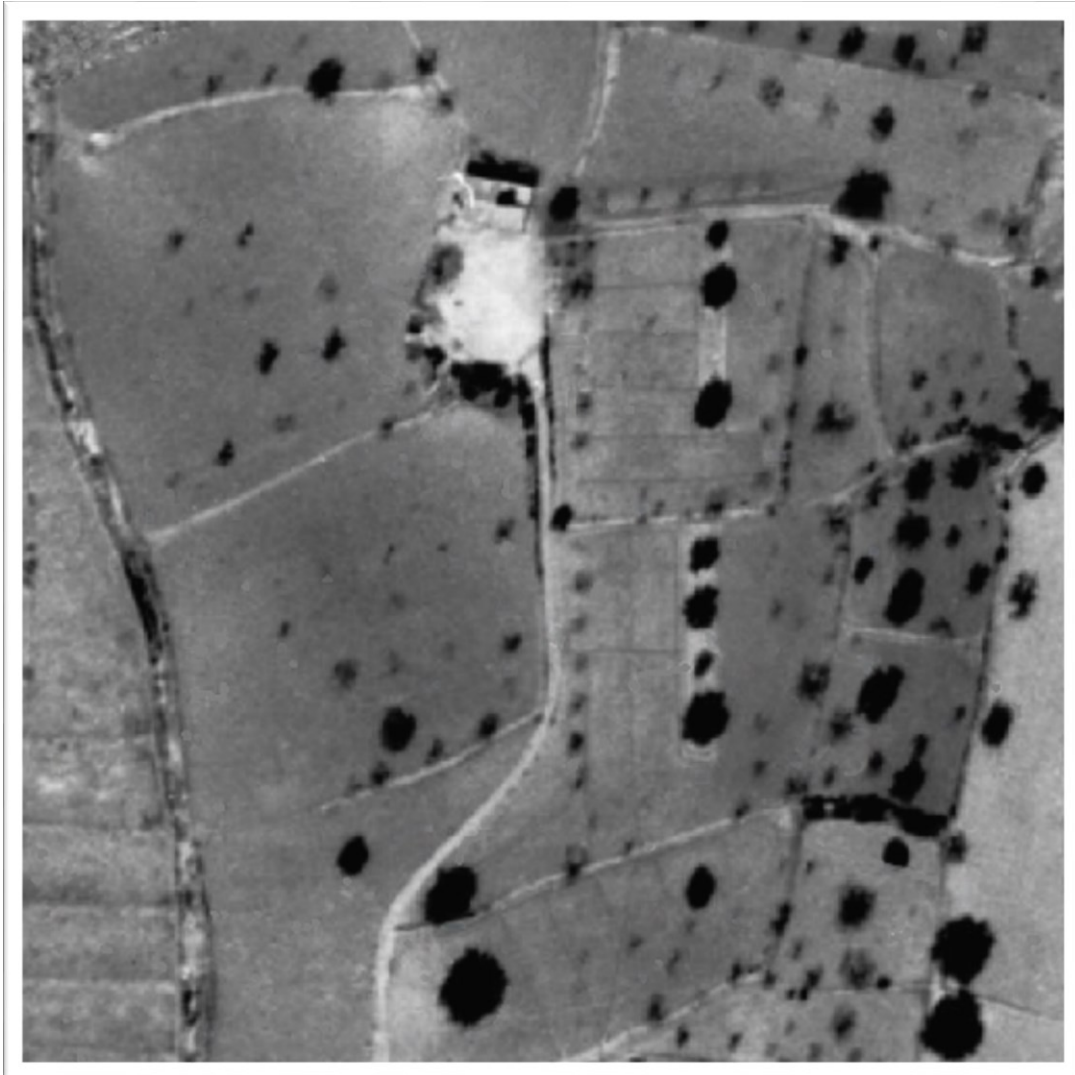


Figura 7.38. Ortofotografía del Vuelo de Ruiz de Alda restaurada con el modelo
DeepMapsRuidoyDesenfoqueBN

Fuente: elaboración propia a partir de los datos de la Confederación Hidrográfica de la Cuenca
del Segura

Como puede apreciarse en la imagen anterior, el uso del Deep Learning en cartografía es capaz de mejorar notablemente la calidad de los datos disponibles hoy en día, y además con un coste mínimo, ya que una vez han sido entrenadas correctamente, las redes neuronales son capaces de realizar las tareas para las que han sido entrenadas de forma automatizada.

7.5.3. Caso 5. Ejemplo 3: Corrección de unión mal realizada entre ortofotografías del Vuelo de Ruiz de Alda en el área de Murcia

El error para corregir en este ejemplo deriva de la unión deficiente entre dos ortofotografías, este tipo de problemas es muy frecuente en cartografías de todo tipo, incluso en las recientes. Todos estos errores derivados de la unión de dos imágenes distintas pueden sin embargo resolverse de forma automatizada con varias de las inteligencias artificiales creadas. Sin embargo, ya que en los ejemplos presentados en este caso el fin es únicamente eliminar los errores y deterioros presentes en la ortofotografía original, preservando todas las características de la imagen suministrada por el IGN, se ha aplicado la red neuronal DeepMapsRuidoyDesenfoqueBN. En otros casos incluidos en este capítulo puede comprobarse cómo algunas de las otras redes creadas durante esta investigación son capaces, no sólo de corregir errores, sino de mejorar y enriquecer los datos originales.

En este ejemplo, el error se encuentra muy próximo al Estadio de la Condomina en Murcia, concretamente en las coordenadas 664993.7514869997, 665170.7048205135, 4206011.060399666, 4206185.685399845 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].

Tanto la ortofotografía original como la corregida, están representadas en este capítulo con una escala de 1:1.000.



Figura 7.39. Ortofotografía original del Vuelo de Ruiz de Alda con polvo y arañazos
Fuente: elaboración propia a partir de los datos de la Confederación Hidrográfica de la Cuenca del Segura

Este ejemplo muestra lo útil que sería la aplicación sistematizada de un procedimiento automático de tratamiento de imágenes basado en inteligencia artificial.

Si se aplicaran los procedimientos propuestos en esta tesis, podría mejorarse en gran medida la calidad de los datos disponibles, tal y como puede apreciarse en los resultados de este caso expuestos a continuación, donde se puede comprobar como no sólo se ha corregido las evidencias de la unión de imágenes, sino todos los deterioros producto de la antigüedad de la ortofotografía.



Figura 7.40. Ortofotografía del Vuelo de Ruiz de Alda restaurada con el modelo
DeepMapsRuidoyDesenfoqueBN
Fuente: elaboración propia a partir de los datos de la Confederación Hidrográfica de la Cuenca
del Segura

Como puede comprobarse con los resultados obtenidos, las redes neuronales que han sido creadas son capaces de corregir de forma automatizada los frecuentes errores presentes en las ortoimágenes, principalmente en las más antiguas, debidos al paso del tiempo o a la falta de celo de aquellos que las almacenaron y escanearon.

7.6. Caso 6: Mejora de la resolución

Para este caso se ha optado por la mejora de la resolución de las ortoimágenes del PNOA de Máxima Actualidad, para la demostración de cómo las inteligencias artificiales generadas pueden aumentar la calidad de los datos más precisos que actualmente se tienen de todo el territorio español.

7.6.1. Caso 6. Ejemplo 1: Campo de cultivo (Comunidad de Madrid)

Para este ejemplo se ha escogido un campo de cultivo situado entre San Sebastián de los Reyes y Daganzo de Arriba, en la Comunidad de Madrid. La imagen en la que se muestra la eficacia del modelo DeepMapsDespixela, se corresponde al área roja mostrada en la siguiente imagen.



Figura 7.41. Localización del área mostrada en el ejemplo
Fuente: elaboración propia a partir de los datos del IGN

Esta área roja de la cual se ha extraído la ortofotografía mostrada a continuación tiene las siguientes coordenadas: 453068.43524177023, 453073.62954177556, 4488998.426919641, 4489003.665669646 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.42. Ortoimagen original del PNOA de Máxima Actualidad
Fuente: elaboración propia a partir de los datos del IGN

La ortoimagen superior se corresponde a los datos originales del PNOA de Máxima Actualidad suministrados por el IGN y al igual que la ortoimagen corregida que se muestra a continuación tiene una escala de 1:30.



Figura 7.43. Ortoimagen generada por la inteligencia artificial DeepMapsDespoxela
Fuente: elaboración propia a partir de los datos del IGN

Como puede apreciarse en el resultado, la inteligencia DeepMapsDespoxela, es capaz de aumentar la resolución de las ortoimágenes, eliminando la pixelación y dando un resultado realista que mantiene la distribución de elementos presentes en la ortoimagen original.

7.6.2. Caso 6. Ejemplo 2. Paso de cebra (Comunidad de Madrid)

En este ejemplo se proporciona una muestra de cómo la inteligencia artificial DeepMapsDespixela es capaz de mejorar la resolución de un área urbana.

La localización de la ortofotografía utilizada en este ejemplo es la que se muestra en la siguiente imagen en el área roja.



Figura 7.44. Localización del área mostrada en el ejemplo
Fuente: elaboración propia a partir de los datos del IGN

Esta área roja de la cual se ha extraído la ortofotografía mostrada a continuación tiene las siguientes coordenadas: 439413.0499090882, 439430.36424243916, 4472457.252057785, 4472474.714557802 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.45. Ortoimagen original del PNOA de Máxima Actualidad
Fuente: elaboración propia a partir de los datos del IGN

La imagen anterior corresponde a un paso de cebrá en el acceso a la Glorieta de las Pirámides por el Paseo de Yaserías (Madrid). Esta imagen procede de los datos originales del PNOA de Máxima Actualidad y tiene una escala de 1:100, al igual que la imagen generada por el modelo, la cual se muestra a continuación.

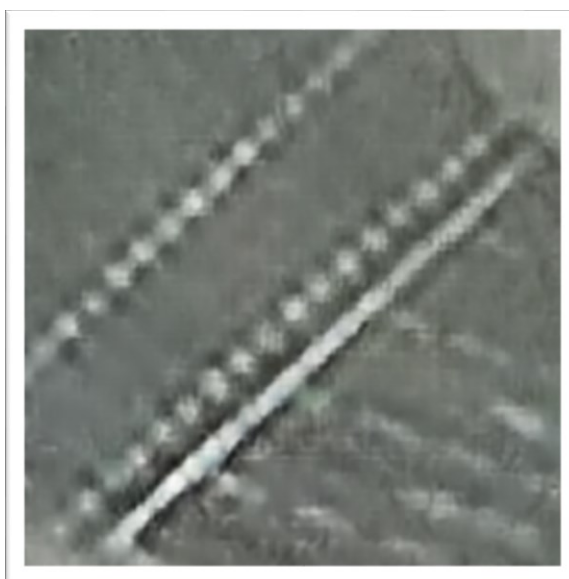


Figura 7.46. Ortoimagen generada por la inteligencia artificial DeepMapsDespoxela
Fuente: elaboración propia a partir de los datos del IGN

Tal y como se aprecia en el resultado, se produce un aumento de nitidez; sin embargo, tal y como puede comprobarse en los distintos ejemplos incluidos en este caso, los mejores resultados se obtienen al aplicar la red neuronal DeepMapsRestaura a imágenes con una escala mayor, ya que ha sido entrenada para mejorar la resolución en escalas mayores.

7.6.3. Caso 6. Ejemplo 3: Carretera (Comunidad de Madrid)

Este ejemplo muestra la mejora de la resolución de la ortoimagen de una sección de la Autovía del Sur de Madrid, cercana a una intersección con la Calle Embajadores, con una escala de 1:125. El emplazamiento de los datos utilizados en este ejemplo viene marcado en la siguiente imagen con un área en color rojo.



Figura 7.47. Localización del área mostrada en el ejemplo
Fuente: elaboración propia a partir de los datos del IGN

El área mostrada en este ejemplo y señalada en color rojo en la imagen anterior, se corresponde con las siguientes coordenadas: 442010.64970340143, 442032.2926200901, 4469640.6407632455, 4469662.468888268 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].

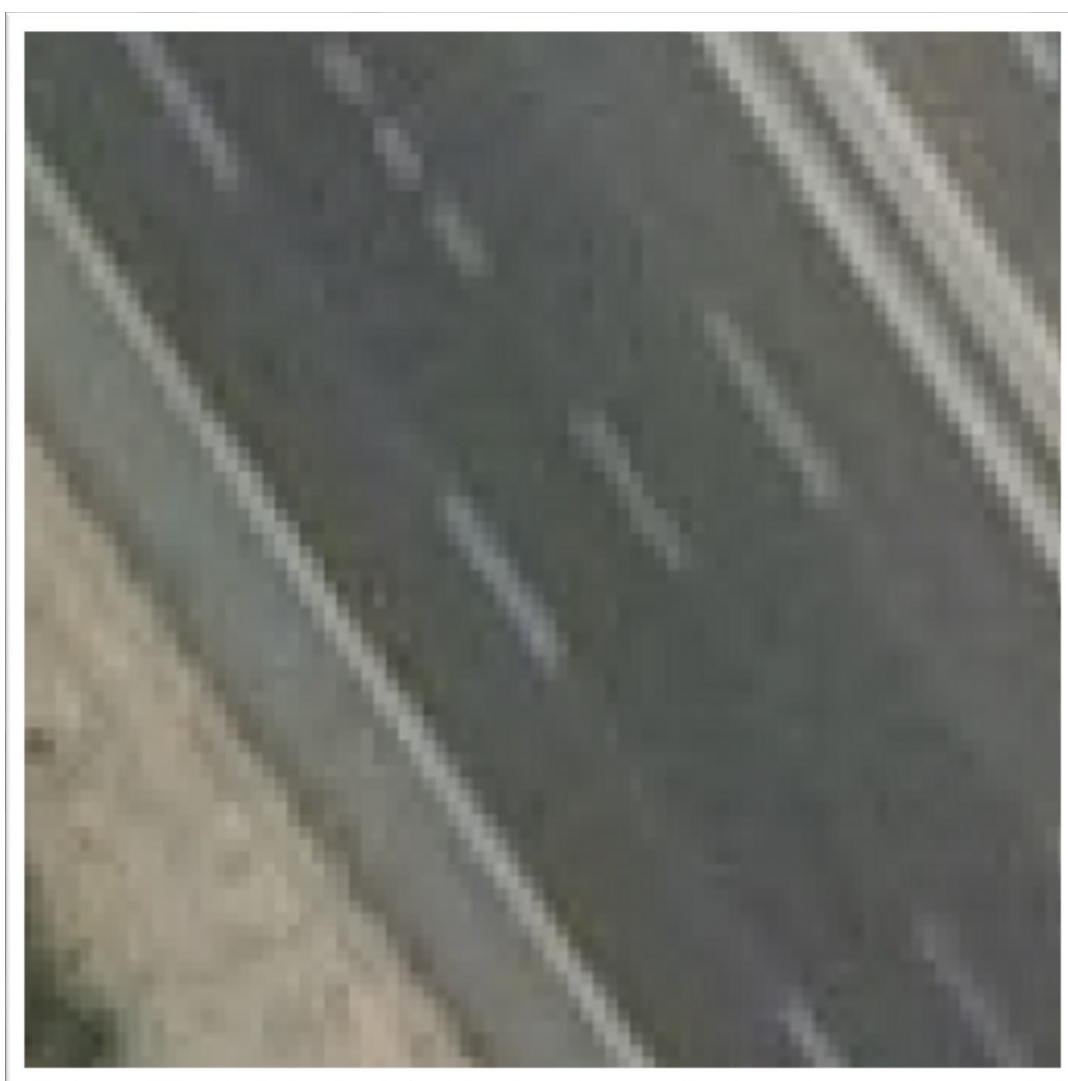


Figura 7.48. Ortoimagen original del PNOA de Máxima Actualidad
Fuente: elaboración propia a partir de los datos del IGN

Tal y como se aprecia en la ortoimagen original del PNOA situada sobre este párrafo, los datos originales ofrecidos por el IGN muestran una pixelación evidente en la escala de 1:125; sin embargo, gracias al modelo DeepMapsDespoxela, se puede mejorar la resolución y eliminar ese pixelado, como se muestra en la siguiente ortoimagen.



Figura 7.49. Ortoimagen generada por la inteligencia artificial DeepMapsDespoxela
Fuente: elaboración propia a partir de los datos del IGN

7.6.4. Caso 6. Ejemplo 4: Embalse (Cantabria)

Este ejemplo muestra cómo, mediante la aplicación de las redes neuronales desarrolladas, puede mejorarse la resolución incluso de elementos poco habituales o inexistentes en los datos de entrenamiento. En este caso de los reflejos presentes en el agua de un embalse situado en Villanueva, de Las Rozas (Cantabria).

La imagen que se muestra en este ejemplo se corresponde con el área marcada en rojo en la siguiente ortoimagen.



Figura 7.50. Localización del área mostrada en el ejemplo
Fuente: elaboración propia a partir de los datos del IGN

Tanto la imagen original como la imagen generada por DeepMapsDespixela tienen una escala de 1:50, y se corresponden con las siguientes coordenadas: 416316.4982477491, 416325.15541442466, 4758522.679134429, 4758531.410384438 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].

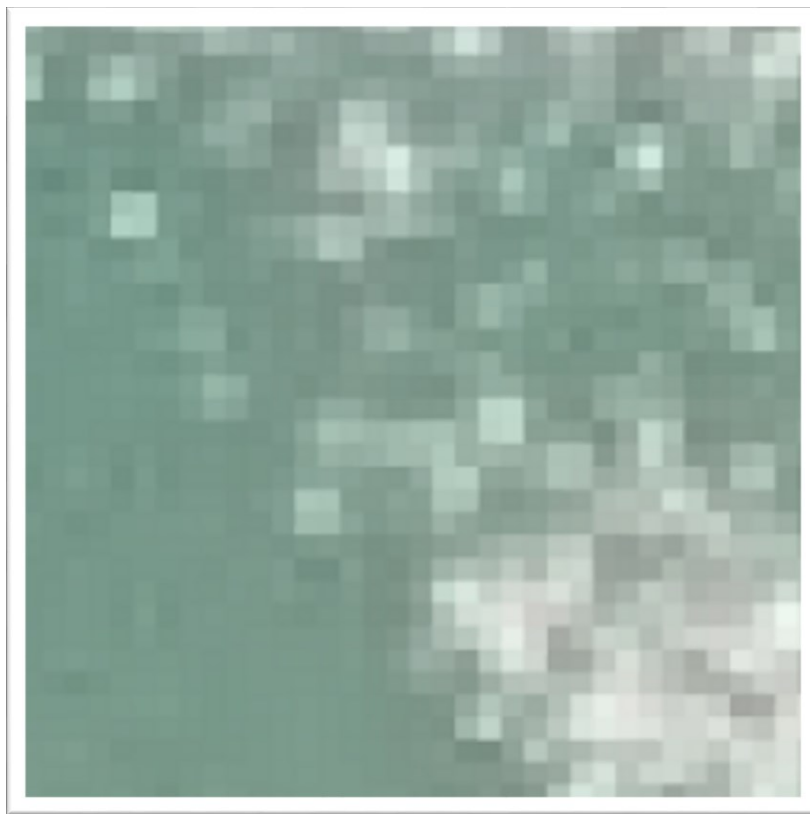


Figura 7.51. Ortoimagen original del PNOA de Máxima Actualidad
Fuente: elaboración propia a partir de los datos del IGN

La imagen superior, que muestra los datos originales del PNOA y la imagen generada por DeepMapsDespixela, representan una pequeña fracción del Embalse del Ebro, concretamente el área situada al norte de la capilla-cementerio de Las Rozas.

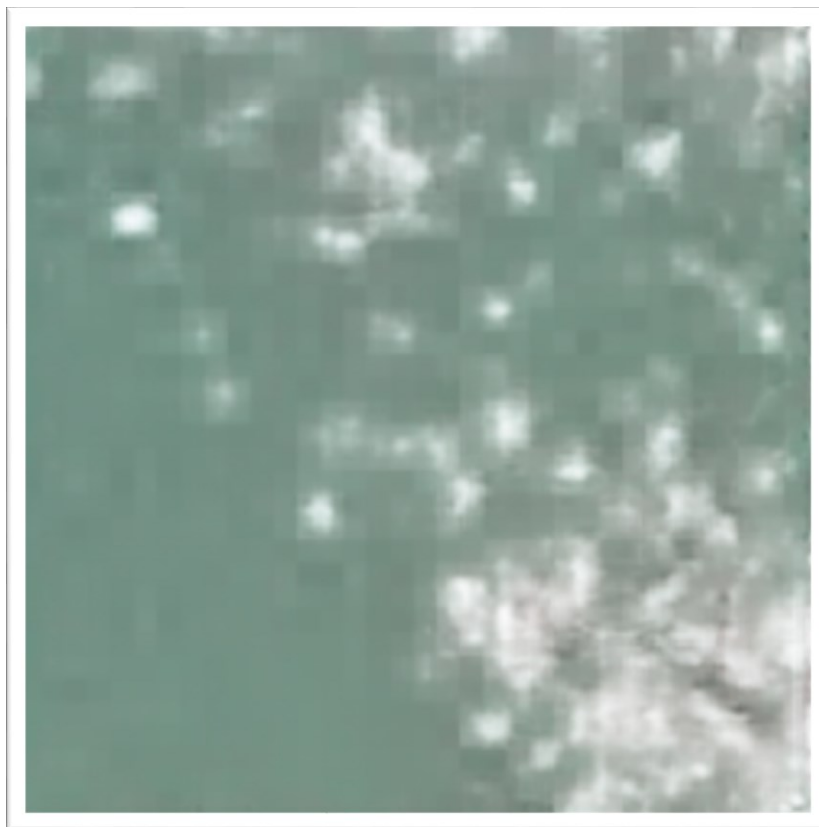


Figura 7.52. Ortoimagen generada por la inteligencia artificial DeepMapsDespoxela
Fuente: elaboración propia a partir de los datos del IGN

Tal y como se muestra en la imagen anterior, el pixelado de los datos originales ha disminuido notablemente y han aumentado tanto la nitidez como el realismo de la ortoimagen.

7.6.5. Caso 6. Ejemplo 5: Parque del Retiro (Comunidad de Madrid)

Este ejemplo muestra el aumento de resolución en un área perteneciente al Parque del Retiro, en la ciudad de Madrid, correspondiente con las siguientes coordenadas: 442010.48250433546, 442013.94537100557, 4473929.751699244, 4473933.244199247 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].

El área que se muestra en este ejemplo es la misma que se muestra en rojo en la siguiente imagen, pero con una escala de 1:20.



Figura 7.53. Localización del área mostrada en el ejemplo
Fuente: elaboración propia a partir de los datos del IGN



Figura 7.54. Ortoimagen original del PNOA de Máxima Actualidad
Fuente: elaboración propia a partir de los datos del IGN

La imagen anterior muestra los datos originales del PNOA con escala de 1:20, la misma que la que se muestra en la imagen posterior, la cual representa los datos generados por la inteligencia artificial DeepMapsDespoxela.



Figura 7.55. Ortoimagen generada por la inteligencia artificial DeepMapsDespoxela
Fuente: elaboración propia a partir de los datos del IGN

Como ha podido comprobarse en los distintos ejemplos de este caso, la red neuronal DeepMapsDespixela ofrece los mejores resultados para escalas comprendidas entre 1:20 y 1:30; cuanto más nos alejamos de esta escala los resultados van perdiendo calidad; sin embargo, suponen una mejoría notable de los datos originales hasta escalas de 1:200.

7.7. Caso 7: Restauración de ortofotografías de la Cuenca del Segura del Vuelo Ruiz de Alda de 1929-1930

Las capacidades de las inteligencias artificiales desarrolladas en esta investigación no se limitan a la restauración de las imágenes antiguas devolviéndolas a su estado original, sino que son capaces de mejorarlas hasta niveles que nunca habían tenido, añadiendo incluso nueva información correspondiente a los datos más probables basados en analítica predictiva. Gracias a ello pueden mejorar la calidad original de las imágenes e, incluso, incluir colores plausibles en aquellas que carecen de ellos debido a su antigüedad.

Para la demostración de la eficacia de las inteligencias artificiales que se han entrenado en este tipo de tareas, se ha optado por aplicarlas en ortoimágenes aéreas antiguas (del Vuelo de Ruiz de Alda en este caso y del Vuelo Americano Serie B en el caso 8, del siguiente epígrafe), las cuales en origen carecían del color y la definición actuales debido a las limitaciones técnicas de la época en la que fueron recogidos los datos.

A pesar de la eficacia que las distintas inteligencias artificiales entrenadas durante este estudio han demostrado para labores concretas, en tareas como la restauración de imágenes se requeriría la aplicación de varias de ellas conjuntamente, lo cual alargaría el tiempo de procesado y supondría un obstáculo para su aplicación de forma masiva, si bien seguirán siendo idóneas para tareas de investigación o análisis en profundidad de datos concretos. Es por ello por lo que, pensando en el tratamiento masivo automatizado de datos, se han creado las inteligencias artificiales que se muestran a continuación (en los casos 7 y 8), las cuales permiten realizar de manera automática todas las tareas necesarias para distintos casos de uso reales, no siendo necesaria la intervención humana en este proceso.

Al objeto de demostrar la posibilidad de creación de una inteligencia artificial capaz de trabajar con datos de fuentes muy diversas y proporcionar buenos resultados con todas ellas, lo cual es necesario para su implantación como herramienta práctica que vaya más allá de los propósitos de investigaciones concretas, se han desarrollado los modelos DeepMapsRestaura15, DeepMapsRestaura17 y DeepMapsRestaura19, estando los detalles de sus entrenamientos recogidos en el capítulo 5.

Estas inteligencias artificiales son capaces de realizar las tareas de coloración, eliminación de ruido, eliminación de desperfectos, corrección del brillo, corrección de contraste, mejora de la resolución, corrección del desenfoque y mejora de la nitidez en general, detectando las deficiencias de las ortoimágenes originales y aplicando las distintas correcciones en el grado necesario en función de los datos a tratar de forma automática, para proporcionar una salida con características similares a las ortoimágenes del PNOA de Máxima Actualidad.

En este caso se han escogido las ortofotografías aéreas más antiguas proporcionadas por el IGN, las cuales tienen una antigüedad superior a los 90 años. Los ejemplos mostrados en este caso pertenecen a la Cuenca del Segura, ya que es la única zona que fue cartografiada en ese vuelo.

7.7.1. Caso 7: Ejemplo 1: El Palmar (Murcia)

La zona mostrada en este ejemplo está situada en las faldas de la sierra de El Puerto, perteneciente a la Sierra de Carrascoy, y junto al canal del Reguerón, por donde discurre el río Guadalentín, en la comarca de la Huerta de Murcia. La pedanía de “El Palmar” recogida en este ejemplo se encuentra en las siguientes coordenadas: 661201.7643164597, 661733.2593170018, 4200597.9499774575, 4201091.725977961 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].

Tanto la imagen original que se muestra a continuación, como la imagen generada tienen una escala de 1:2.700.



Figura 7.56. Ortofotografía original del Vuelo Ruiz de Alda
Fuente: elaboración propia a partir de los datos del IGN

En la siguiente imagen se puede apreciar cómo la inteligencia artificial DeepMapsRestaura17 ha asignado color, corregido el enfoque y mejorado la calidad general de la ortoimagen original.



Figura 7.57. Ortoimagen generada por la inteligencia artificial DeepMapsRestarua17
Fuente: elaboración propia a partir de los datos del IGN

7.7.2. Caso 7: Ejemplo 2: Campos de labor (Murcia)

En el ejemplo se muestran campos de labor de Murcia, que actualmente forman parte de la ciudad, concretamente la parte correspondiente a la Ronda Circular, la Avenida de Don Juan de Borbón y la Ronda de Levante en la ciudad de Murcia.



Figura 7.58. Ortofotografía original del Vuelo Ruiz de Alda
Fuente: elaboración propia a partir de los datos del IGN

Las coordenadas tanto de la ortoimagen original del Vuelo Ruiz de Alda, como de la ortoimagen corregida son las siguientes: 664298.7404853104, 664945.1704859696, 4206199.535000731, 4206738.01500128 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)]. La escala de las imágenes de este ejemplo es de 1:15.000. El archivo del cual ha sido extraída la ortoimagen mostrada en este ejemplo es la hoja m127_120_geo.ecw del IGN.



Figura 7.59. Ortoimagen generada por la inteligencia artificial DeepMapsRestarua15
Fuente: elaboración propia a partir de los datos del IGN

Como puede verse en la ortoimagen anterior, la inteligencia artificial DeepMapsRestaura15, ha asignado color, corregido el enfoque, la luminosidad, el contraste y, en definitiva, mejorado la calidad general de la ortoimagen original.

7.7.3. Caso 7: Ejemplo 3: Plaza de Toros y Estadio de la Condomina (Murcia)

El área recogida en este ejemplo contiene dos elementos destacados de la ciudad de Murcia, los cuales aún permanecen en la actualidad, éstos son la Plaza de Toros de la ciudad y el estado de fútbol de la Condomina.

La siguiente ortoimagen muestra el área comprendida en las coordenadas: 664817.353778481, 665112.6287787822, 4205755.181857716, 4206029.501857996 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].

La escala tanto de la ortoimagen original del IGN como de la imagen generada por el modelo DeepMapsRestaura19 es de 1:1.500.



Figura 7.60. Ortofotografía original del Vuelo Ruiz de Alda
Fuente: elaboración propia a partir de los datos del IGN

Como puede verse en la siguiente ortoimagen la inteligencia artificial DeepMapsRestaura19, ha corregido varias de las deficiencias de la ortoimagen original, generando un resultado mucho más cercano a las ortoimágenes actuales.

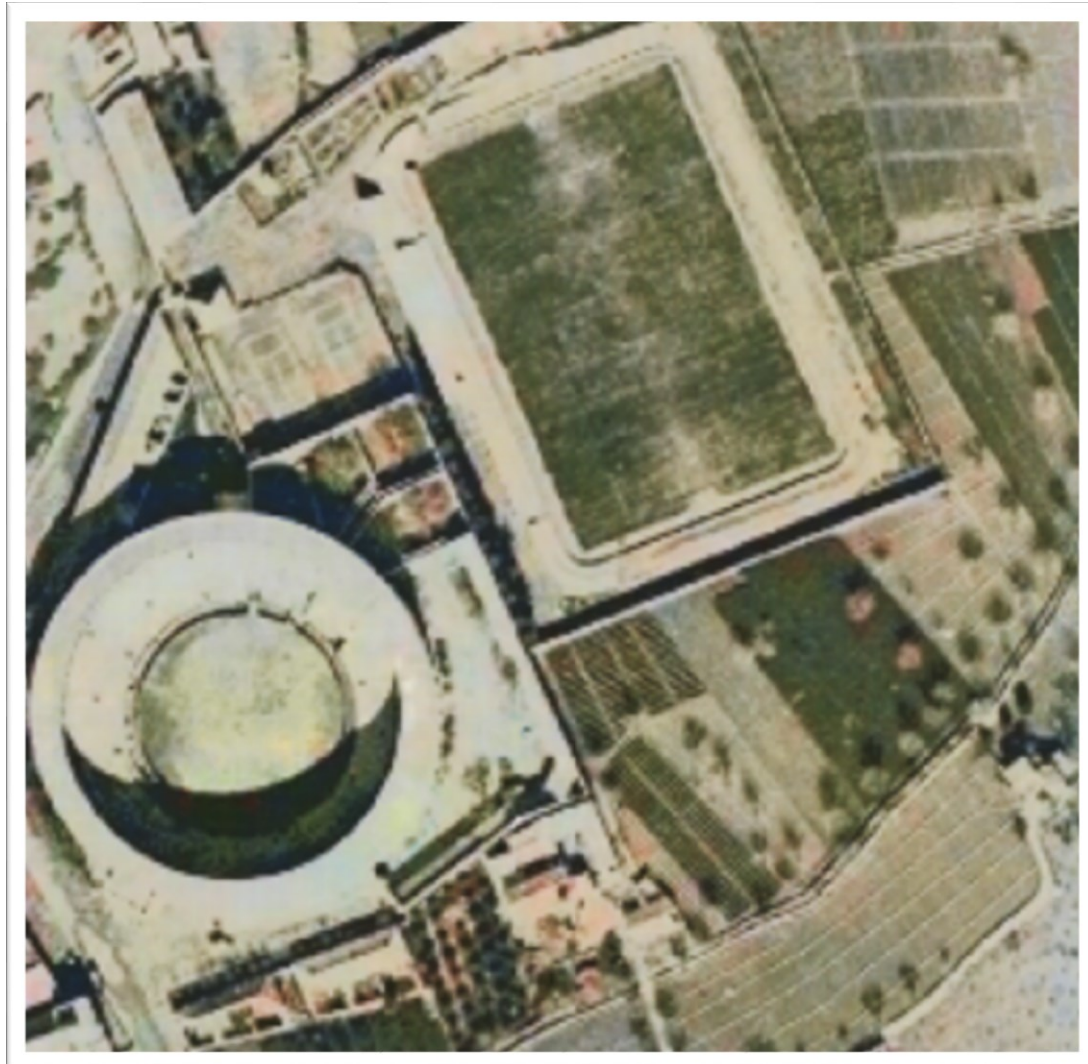


Figura 7.61. Ortoimagen generada por la inteligencia artificial DeepMapsRestarua19
Fuente: elaboración propia a partir de los datos del IGN

7.7.4. Caso 7: Ejemplo 4: Elche (Alicante)

En este ejemplo muestra una parte de la ciudad de Elche, comprendida en las siguientes coordenadas: 700851.4395235727, 701560.0995242955, 4237475.257515093, 4238133.625515765 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)]. Esta zona incluye la Plaza de España, la antigua Plaza de Toros y Pont de Canalejas. Tanto la ortofotografía aérea original como la restaurada por la inteligencia artificial DeepMapsRestaura15 tienen una escala de 1:10.000.



Figura 7.62. Ortofotografía original del Vuelo Ruiz de Alda
Fuente: elaboración propia a partir de los datos del IGN

Como puede verse, en la siguiente imagen la inteligencia artificial DeepMapsRestaura15 ha restaurado la ortoimagen original, corrigiendo varias de las deficiencias de ésta, acercándola de esta manera a las ortoimágenes actuales.



Figura 7.63. Ortoimagen generada por la inteligencia artificial DeepMapsRestarua15
Fuente: elaboración propia a partir de los datos del IGN

7.8. Caso 8: Restauración de ortofotografías de Madrid del Vuelo Americano (Serie B, 1956-1957)

En este caso se exponen varios ejemplos de la aplicación de algunas de las inteligencias artificiales creadas durante la presente investigación sobre ortofotografías del Vuelo Americano Serie B. En ellas puede comprobarse cómo las redes neuronales son capaces de mejorar la calidad de las ortofotografías y asignarles colores verosímiles a sus elementos, algo que hasta esta tesis nunca se había llevado a cabo con ortoimágenes.

Los datos de partida con los que han sido alimentadas las inteligencias artificiales que han generado las imágenes restauradas y en color, de las cuales se han extraído las ortoimágenes que se muestran como ejemplos, proceden del servicio WMS del IGN [“https://www.ign.es/wms/pnoa-historico?SERVICE=WMS&”](https://www.ign.es/wms/pnoa-historico?SERVICE=WMS&), del cual se ha utilizado la capa AMS_1956-1957, que tiene por título “Vuelo americano (Serie B, 1956-1957)”. Esta capa recoge las ortoimágenes del vuelo realizado por el Army Map Service de EEUU entre enero de 1956 y noviembre de 1957 sobre parte del territorio español, conocido como Vuelo Americano, serie B.

En este caso, tanto las ortoimágenes originales como las generadas por las inteligencias artificiales tienen como sistema de referencia de coordenadas el EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado).

7.8.1. Caso 8: Ejemplo 1: Jardines del Campo del Moro (Madrid)

Este ejemplo muestra la restauración de la ortofotografía procedente del vuelo americano serie B de los Jardines del Campo del Moro, en la cual se

aprecian, además de estos jardines (centro de la imagen), el Río Manzanares (izquierda), el Palacio Real y la Catedral de la Almudena (derecha). Esta restauración ha sido realizada por la inteligencia artificial DeepMapsRestaura17



Figura 7.64. Ortofotografía original del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

Tanto la ortoimagen original, como la generada por la red neuronal tienen una escala de 1:8.000 y están comprendidas en las siguientes coordenadas: 438546.447647004, 439484.885147004, 4473900.064716404, 4474835.255027823 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.65. Ortoimagen generada por la inteligencia artificial DeepMapsRestaura17 a partir de los datos del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

Tal y como puede apreciarse en la ortoimagen anterior, la inteligencia artificial DeepMapsRestaura17, ha asignado color, corregido el enfoque y mejorado la calidad general de la ortoimagen original.

7.8.2. Caso 8: Ejemplo 2: Plaza de Moncloa (Madrid)

En este ejemplo puede apreciarse cómo la red neuronal DeepMapsRestaura17 ha añadido color a una ortofotografía del Vuelo

Americano Serie B, en la que pueden verse la Plaza de Moncloa y el Cuartel General del Ejército del Aire, habiendo sido también mejorada la calidad.



Figura 7.66. Ortofotografía original del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

Tanto la ortoimagen original, como la generada por la red neuronal están representadas en una escala de 1:8.000 y ambas tienen las siguientes coordenadas: 438550.2360336822, 439488.6735336822, 4475726.608293391, 4476661.79860481 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.67. Ortoimagen generada por la inteligencia artificial DeepMapsRestaura15 a partir de los datos del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

La imagen superior muestra como la red neuronal DeepMapsRestaura15 ha corregido pequeñas imperfecciones de la imagen original y asignado correctamente color tanto a edificios, como a carreteras y vegetación.

7.8.3. Caso 8: Ejemplo 3: Plaza de Colón (Madrid)

Este ejemplo muestra la restauración realizada mediante el uso de la red neuronal DeepMapsRestaura17 de una ortofotografía del Vuelo Americano

Serie B, en la que pueden verse la Plaza de Colón, la Biblioteca Nacional y el Museo Arqueológico Nacional.

Tanto la ortoimagen original, como la generada por la red neuronal están representadas en una escala de 1:8.000 y ambas tienen las siguientes coordenadas siguientes coordenadas: 441338.2948301339, 442354.29483117023, 4474822.436353212, 4475700.260354107 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.68. Ortofotografía original del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

Como puede verse en la siguiente imagen, la inteligencia artificial DeepMapsRestaura17, ha asignado color, corregido el enfoque y mejorado la calidad general de la ortoimagen original.

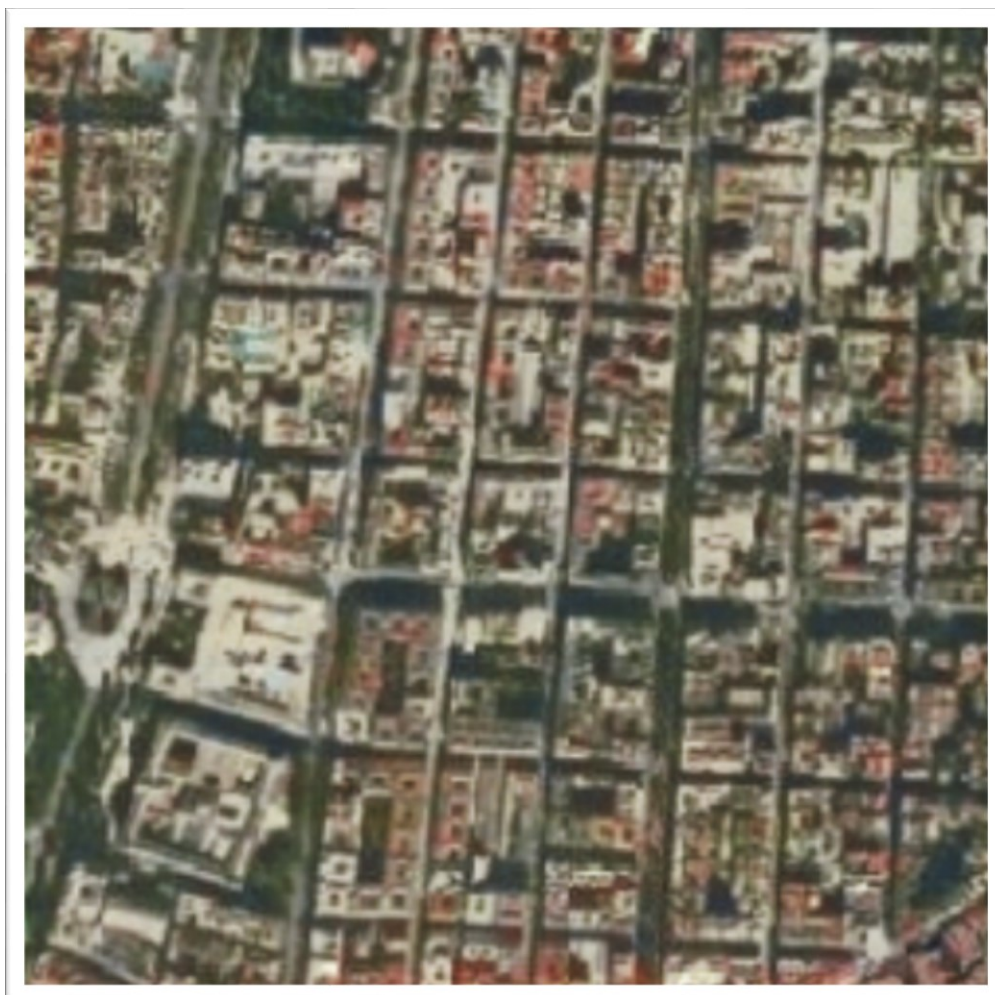


Figura 7.69. Ortoimagen generada por la inteligencia artificial DeepMapsRestarua17 a partir de los datos del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

7.8.4. Caso 8: Ejemplo 4: Paseo del Prado (Madrid)

Este ejemplo muestra como la inteligencia artificial DeepMapsRestarua15 ha incluido color a la ortofotografía de los años 50 de la Calle de Alcalá y el Paseo del Prado a su paso por Cibeles y Neptuno, al tiempo que ha mejorado la calidad de esta.

Tanto la ortofotografía original, como la generada por la inteligencia artificial, tienen una escala de 1:5.000 y muestran el área delimitada por las siguientes coordenadas: 440450.4930761699, 441336.44507707335, 4473927.022949391, 4474804.846950286 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.70. Ortofotografía original del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

En la imagen mostrada a continuación puede comprobarse cómo DeepMapsRestaura17 ha añadido color y mejorado la calidad de la ortofotografía original del Vuelo Americano Serie B.



Figura 7.71. Ortoimagen generada por la inteligencia artificial DeepMapsRestarua15 a partir de los datos del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

7.8.5. Caso 8: Ejemplo 5: Atocha (Madrid)

En este ejemplo puede verse como DeepMapsRestarua15 ha mejorado varios aspectos de la ortofotografía original del Vuelo Americano Serie B en la zona de la Calle de Atocha (Madrid).



Figura 7.72. Ortofotografía original del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

Tanto la ortoimagen original como la generada por la inteligencia artificial corresponden a las siguientes coordenadas: 440445.921076165, 441331.87307706865, 4473002.462948449, 4473880.286949344 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)]. Ambas ortoimágenes tienen una escala de 1:15.000.



Figura 7.73. Ortoimagen generada por la inteligencia artificial DeepMapsRestarua15 a partir de los datos del Vuelo Americano Serie B

Fuente: elaboración propia a partir de los datos del IGN

La ortofotografía aérea restaurada por la inteligencia artificial muestra la Calle de Atocha, el Paseo del Prado, Lavapiés (izquierda) y Colegio de los Salesianos (abajo), en la ciudad de Madrid.

7.8.6. Caso 8: Ejemplo 6: Plaza de Manuel Becerra (Madrid)

Las ortoimágenes mostradas en este ejemplo ilustran cómo las inteligencias artificiales que se han desarrollado son capaces de aprender de forma autónoma a detectar defectos para los que no habían sido entrenadas. Este ejemplo es otra muestra de autoemergencia de conocimiento y nuevos comportamientos en las inteligencias artificiales cuando son entrenadas lo suficiente. En este caso la red neuronal DeepMapsRestaura15 demuestra que es capaz de percibir el contenido de la imagen a nivel de conceptos, para después hacer una representación, desechando elementos que no deberían estar presentes, como el pelo situado en el margen superior derecho en la ortofotografía original del Vuelo Americano Serie B, que se muestra a continuación.



Figura 7.74. Ortofotografía original del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

Las ortoimágenes mostradas en este ejemplo incluyen la Plaza de Manuel Becerra (izquierda) y terreno que actualmente ocupa la M30 (derecha), en la ciudad de Madrid.

Tanto la ortoimagen original, como la generada por la red neuronal están representadas en una escala de 1:15.000 y ambas tienen las siguientes coordenadas: 443205.9039153722, 444221.90391640854, 4474776.790436494, 4475654.614437389 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.75. Ortoimagen generada por la inteligencia artificial DeepMapsRestaura15 a partir de los datos del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

En la ortoimagen anterior puede apreciarse cómo además de añadir color a la ortofotografía original y mejorar el enfoque y la resolución, la inteligencia artificial DeepMapsRestaura15 es capaz de eliminar un elemento ajeno al contexto (un pelo presente durante el proceso de escaneado de las imágenes), a pesar de que la red DeepMapsRestaura15 no ha sido entrenada expresamente para la eliminación de elementos externos. Cuando se entrena una red neuronal cartográfica lo suficiente, esta es capaz, de forma autónoma, de reconocer elementos extraños y eliminarlos.

7.8.7. Caso 8: Ejemplo 7: Puerta de Alcalá (Madrid)

En este caso la inteligencia artificial DeepMapsRestaura19 ha mejorado claramente los datos originales del vuelo americano serie B correspondientes a la Puerta de Alcalá (esquina superior izquierda) y parte del Retiro (derecha), de la ciudad de Madrid.



Figura 7.76. Ortofotografía original del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

Tanto la ortoimagen original, como la generada por la red neuronal están representadas en una escala de 1:3.000 y ambas tienen las siguientes coordenadas: 441496.0605802944, 441813.5605806183, 4474388.6149361, 4474662.93493638 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)]



Figura 7.77. Ortoimagen generada por la inteligencia artificial DeepMapsRestaura19 a partir de los datos del Vuelo Americano Serie B a partir de los datos del Vuelo Americano Serie B
Fuente: elaboración propia a partir de los datos del IGN

En este ejemplo puede comprobarse que cuanto mayor es la escala de la ortofotografía, más sencillo le resulta a la inteligencia artificial identificar los elementos individuales que aparecen en la misma, tal y como se había ya observado en capítulos anteriores.

7.9. Caso 9 Generación automatizada de mapas de usos del suelo con categorías de SIOSE

En este caso se muestran algunos ejemplos de mapas de usos del suelo generados por la inteligencia artificial DeepMapsUsosSIOSE. Durante esta investigación se han desarrollado redes neuronales capaces de generar mapas de usos de suelo con características de distintas fuentes. Para demostrar la capacidad de las inteligencias artificiales a la hora de generar mapas de usos en distintas escalas además del modelo DeepMapsUsosNatura que ha sido creado para generar mapas de precisos a escalas cercanas a 1:4.000, se ha entrenado el modelo DeepMapsUsosSIOSE, el cual está orientado a escalas menores cercanas a 1:70.000 y el modelo DeepMapsSuperficies para escalas cercanas a 1:500.000.

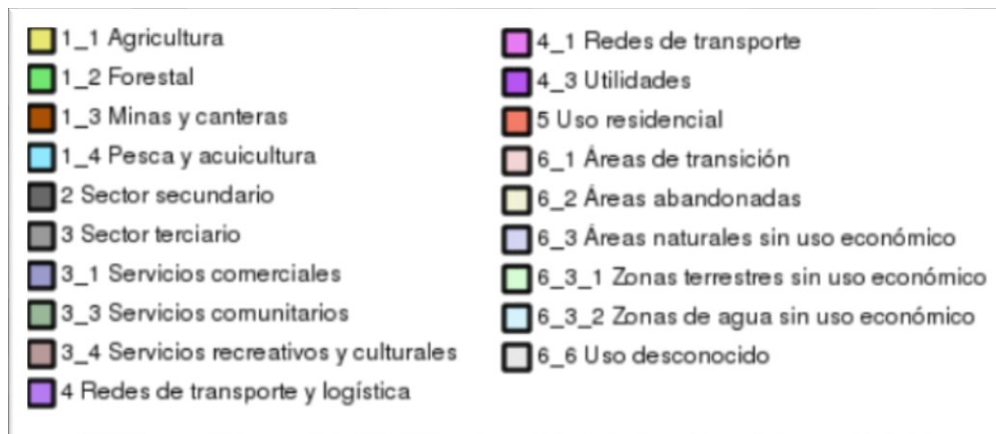


Figura 7.78. Leyenda de los mapas generados por el modelo DeepMapsUsosSIOSE
Fuente: elaboración propia a partir de los datos del geoportal de IDEE

7.9.1. Caso 9. Ejemplo 1: Mapa de usos del suelo de Algete (Madrid)

Este ejemplo muestra un mapa de usos del suelo con las características del SIOSE generado automáticamente por la inteligencia artificial DeepMapsUsosSIOSE a partir de una ortofotografía satelital del Sentinel 2.

El área representada incluye el municipio de Algete (Madrid) y algunos campos de cultivo colindantes. La escala a la que están representadas las siguientes ortoimágenes es de 1:75.000 y sus coordenadas son las siguientes: 456056.8069483555, 459897.5235195353, 4492852.921907851, 4495948.772234923 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.79. Ortoimagen generada por la inteligencia artificial DeepMapsUsosSIOSE a partir de los datos de Sentinel 2

Fuente: elaboración propia a partir de los datos del IGN

7.9.2. Caso 9. Ejemplo 2: Mapa de usos del suelo de Meco (Madrid)

En este ejemplo, al igual que en el anterior, se muestra un mapa de usos del suelo generado sin necesidad de intervención humana por DeepMapsUsosSIOSE, a partir de los datos de Sentinel 2.

El área representada mediante un mapa de usos del suelo de características similares a los mapas del SIOSE, incluye los municipios de Meco y Villanueva de la Torre (esquina superior derecha), ambos pertenecientes a la comunidad de Madrid. Esta área se encuentra comprendida en las siguientes coordenadas: 471268.37227724056, 474678.4630510456, 4488552.483201636, 4492078.959326083 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.80. Ortoimagen generada por la inteligencia artificial DeepMapsUsosSIOSE a partir de los datos de Sentinel 2

Fuente: elaboración propia a partir de los datos del IGN

Tanto la ortofotografía del Sentinel 2 como el mapa de usos generado por la inteligencia artificial tienen una escala de 1:75.000.

7.9.3. Caso 9. Ejemplo 3: Mapa de usos del suelo de la Cuenca del Henares (Madrid)

El mapa de usos del suelo generado en este ejemplo muestra parte de la Cuenca del Henares, al Este de Alcalá de Henares. En la ortofotografía original del Sentinel 2 a partir de la cual DeepMapsUsosSIOSE ha generado su mapa de usos del suelo incluye a la izquierda parte del Parque Natural “Los Cerros”, el cual el modelo DeepMapsUsosSIOSE ha categorizado correctamente al igual que los campos de cultivo y las zonas sin uso económico. El área utilizada en este ejemplo tiene las siguientes coordenadas: 471099.65229660354, 474762.09633688367, 4477479.1650796505, 4480672.527617235 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)]. Las siguientes ortoimágenes tienen una escala de 1:75.000.

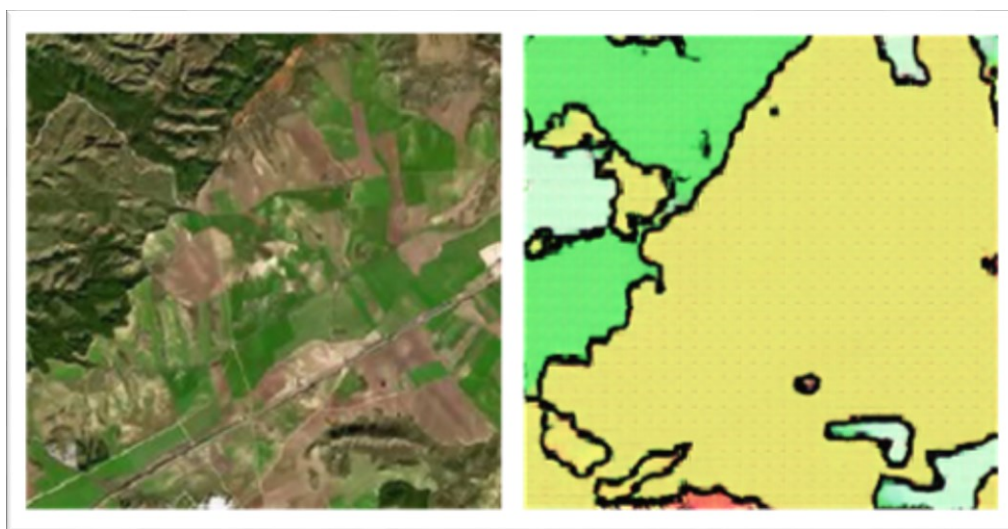


Figura 7.81. Ortoimagen generada por la inteligencia artificial DeepMapsUsosSIOSE a partir de los datos de Sentinel 2

Fuente: elaboración propia a partir de los datos del IGN

7.10. Caso 10: Mapas de densidad de cobertura de árboles

Para los ejemplos de este caso se ha optado por zonas muy distantes de las zonas con cuyos datos fue entrenada la red neuronal DeepMapsBosques, que aquí concretamente ha sido utilizada para la elaboración de los mapas de densidad de cobertura de árboles.

Esta red neuronal fue entrenada con datos procedentes del Copernicus Land Monitoring Service, CORINE y del PNOA de Máxima Actualidad de la Península Ibérica, con un SRC de EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado). Sin embargo, para demostrar la versatilidad de las redes creadas, los datos que se van a procesar mediante esta inteligencia artificial proceden de la capa de ortofotografías aéreas Esri Aéreo, con un SRC de EPSG: 102100 y extraídos de Tasmania, concretamente del parque South Western National Park. A los datos de la capa de Esri Aéreo puede accederse mediante el servicio de mapas

“https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}”. Para todos los ejemplos de este caso se ha escogido la escala 1:28.000.

La leyenda de la capa utilizada para el entrenamiento (HRLForestTCD2015), así como la de los mapas generados por la inteligencia artificial DeepMapsBosques es la que se muestra a continuación.

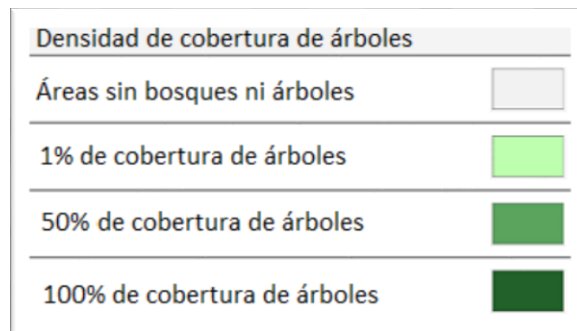


Figura 7.82. Leyenda del modelo DeepMapsBosques

Fuente: elaboración propia a partir de los datos de la leyenda de la capa HRLForestTCD2015 del geoportal de IDEE

Todos los ejemplos mostrados en este caso muestran cómo puede extenderse el conocimiento generado por los humanos en un área concreta, como es el caso de la limitación del Copernicus Land Monitoring Service y del CORINE, los cuales están limitadas a Europa, a cualquier lugar del planeta mediante el uso de la inteligencia artificial.

7.10.1. Caso Ejemplo 1: Densidad de árboles en South Western National Park (Tasmania). Zona1

En este ejemplo las imágenes mostradas a continuación corresponden a la zona del South Western National Park de Tasmania comprendida en las siguientes coordenadas: 16256106.923554784, 16260913.450226353, -5327392.4136043405, -5322502.913599352 [EPSG: 102100 WGS 1984 Web Mercator (Esfera Auxiliar)].

La imagen de la izquierda es una ortofotografía obtenida a partir de la capa “Esri Aéreo”, mientras que la imagen de la derecha sido generada, de forma automática, por la inteligencia artificial DeepMapsBosques partiendo de la ortofotografía.

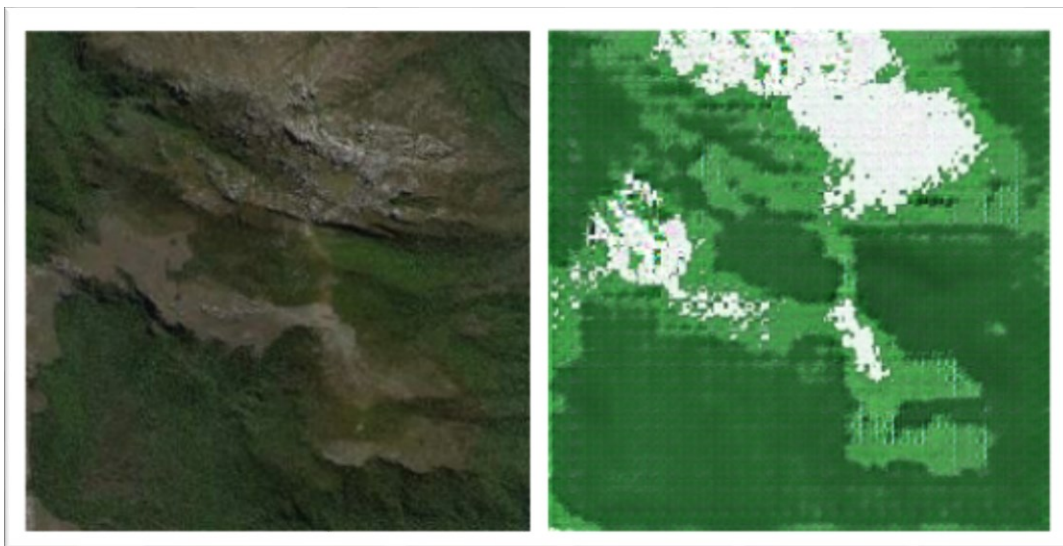


Figura 7.83. Ortoimagen generada por la inteligencia artificial DeepMapsBosques a partir de ortοimagen de la capa Esri Aéreo

Fuente: elaboración propia a partir de los datos de Esri

7.10.2. Caso Ejemplo 2: Densidad de árboles en South Western National Park (Tasmania). Zona2

En este ejemplo, al igual que ocurría en el anterior, se muestran tanto una ortofotografía aérea como el mapa de densidad de árboles generado por la inteligencia artificial DeepMapsBosques a partir de la ortofotografía de la izquierda.

Tanto la ortofotografía aérea como el mapa de densidad de árboles que se muestran a continuación corresponden a la zona del South Western National Park de Tasmania comprendida en las siguientes coordenadas: 16246321.996878149, 16251128.523549719, -5327333.146937617, -5322443.646932629 [EPSG: 102100]

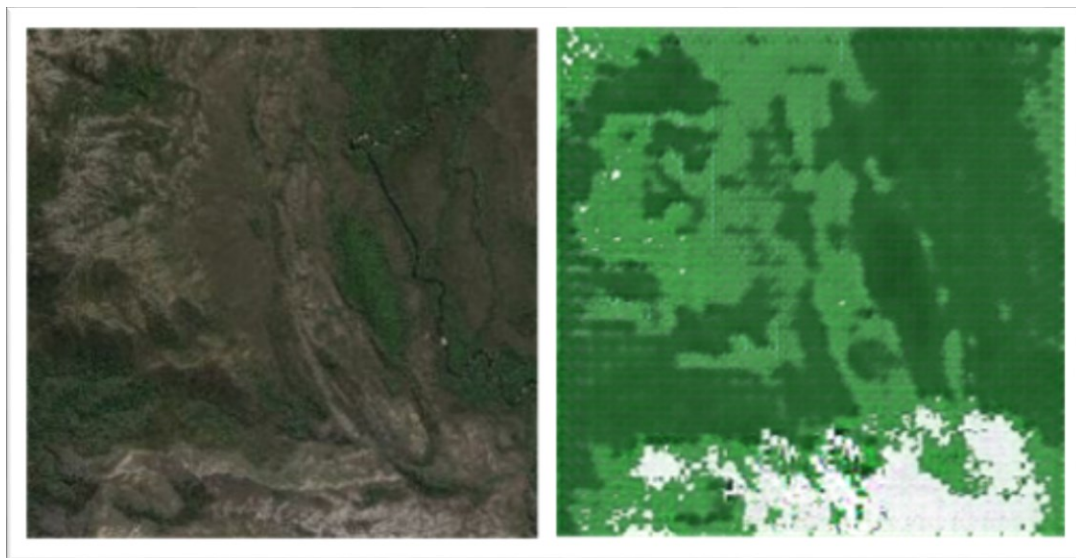


Figura 7.84. Ortoimagen generada por la inteligencia artificial DeepMapsBosques a partir de ortomagen de la capa Esri Aéreo
Fuente: elaboración propia a partir de los datos de Esri

7.10.3. Caso Ejemplo 3: Densidad de árboles en South Western National Park (Tasmania). Zona3

Al igual que en los ejemplos anteriores de este caso, este ejemplo muestra el mapa de densidad de árboles generado por la inteligencia artificial DeepMapsBosques a partir de ortoimágenes de la capa Esri Aéreo correspondientes a un área que forma parte del South Western National Park de Tasmania. En este ejemplo se ha elegido el área correspondiente a las siguientes coordenadas: 16256059.5102214, 16260866.03689297, -5332210.793609258, -5327321.29360427 [EPSG: 102100].

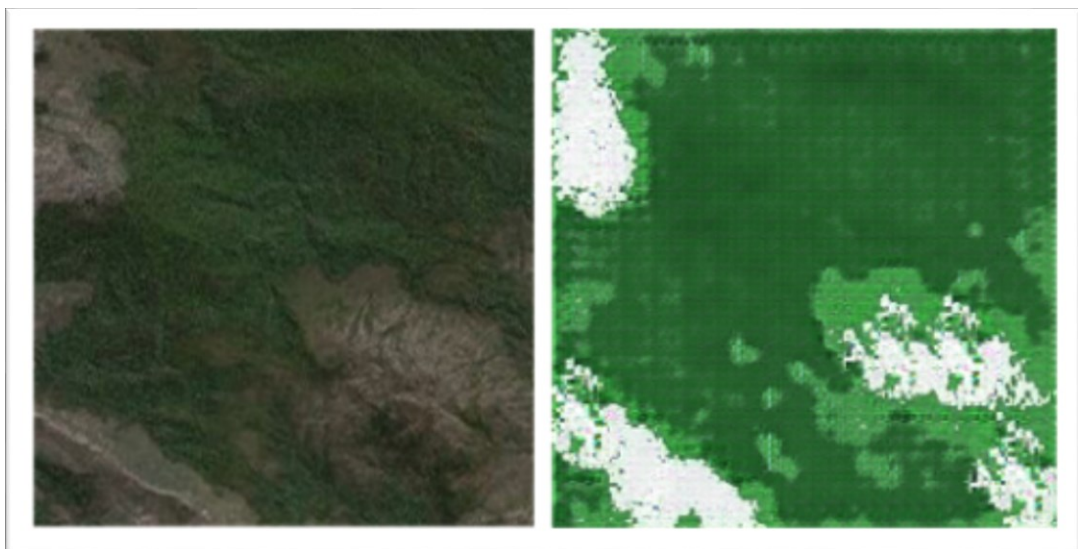


Figura 7.85. Ortoimagen generada por la inteligencia artificial DeepMapsBosques a partir de ortoimagen de la capa Esri Aéreo
Fuente: elaboración propia a partir de los datos de Esri

Como puede comprobarse en las imágenes mostradas anteriormente, la inteligencia artificial aquí desarrollada es capaz de identificar correctamente a partir de ortofotografías, tanto la densidad de árboles existente en las áreas boscosas, como aquellas áreas desprovistas de árboles.

7.10.4. Caso Ejemplo 4: Densidad de árboles en South Western National Park (Tasmania). Zona4

La zona mapeada por DeepMapsBosques en este ejemplo muestra el área con las coordenadas: 16246316.070211474, 16251122.596883044, -5332246.353609296, -5327356.853604307 [EPSG: 102100], situado dentro del South Western National Park de Tasmania. Al igual que el resto de los ejemplos de este caso, la escala tanto del mapa de densidad de árboles generado, como de la ortoimagen de Esri Aéreo a partir de la cual ha sido creado, tienen una escala de 1:28.000.

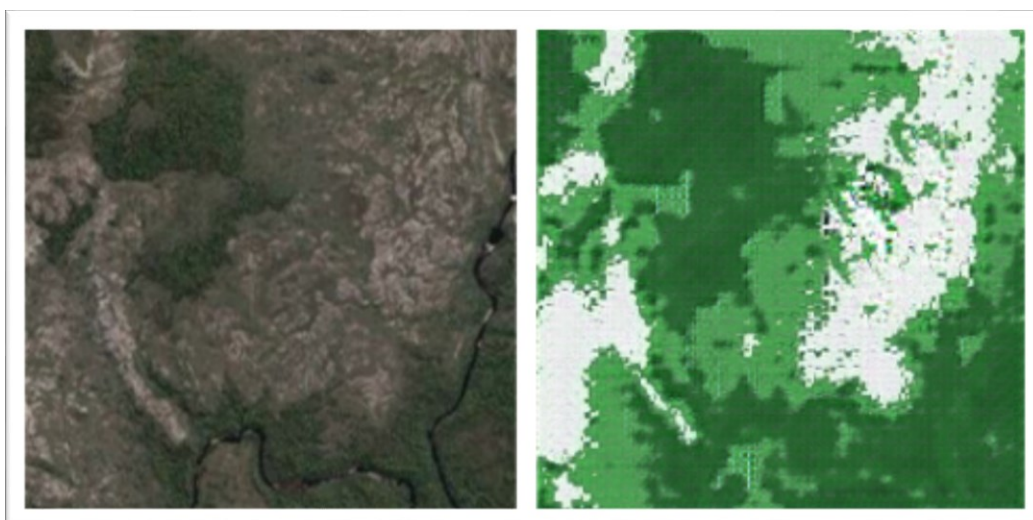


Figura 7.86. Ortoimagen generada por la inteligencia artificial DeepMapsBosques a partir de ortoimagen de la capa Esri Aéreo

Fuente: elaboración propia a partir de los datos de Esri

Los ejemplos mostrados en este caso vienen a demostrar las enormes posibilidades de aplicación de esta inteligencia artificial (DeepMapsBosques), puesto que es potencialmente aplicable a todo el planeta partiendo de ortoimágenes aéreas o satelitales de diversas fuentes, sin necesidad de realizar una adaptación de dominio en la mayoría de los casos (únicamente sería necesaria esta adaptación de dominio en algunas ortofotografías antiguas en blanco y negro, que tengan mala calidad).

7.11. Caso 11: Elaboración de mapas de uso de suelo a partir de ortoimágenes satelitales de menor escala

En este caso se muestran ejemplos de mapas de uso generados de forma automática por la inteligencia artificial DeepMapsSuperficies a partir de ortofotografías satelitales de Sentinel 2.

Las categorías de usos del suelo representadas en los ejemplos de este caso se corresponden a las categorías existentes en la capa “Usos de suelo existentes”, creada a partir de los datos del CORINE y del SIOSE, la cual está disponible a través del siguiente WMS “<https://servicios.idee.es/wms-inspire/ocupacion-suelo?SERVICE=WMS&>”, y cuya leyenda es la siguiente:

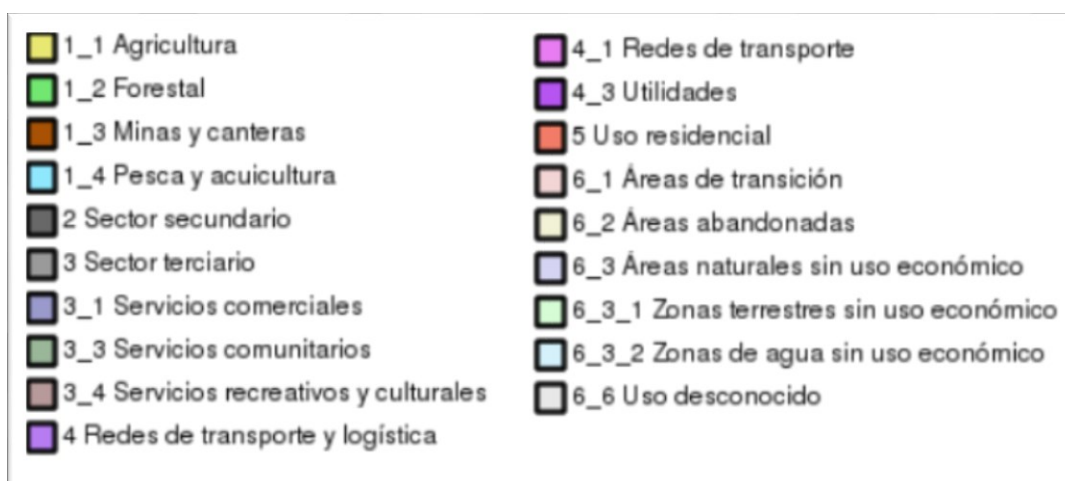


Figura 7.87. Leyenda de los mapas generados por el modelo DeepMapsUsosSIOSE
Fuente: elaboración propia a partir de los datos del geoportal de IDEE

Esta red neuronal ha sido creada para trabajar con escalas cercanas a 1:500.000, la cual coincide con ejemplos que se muestran a continuación.

7.11.1. Caso 11: Ejemplo 1: Mapa de usos del suelo del área entre Lerma y Aranda de Duero

Este ejemplo muestra el mapa de usos de suelo con el estilo del SIOSE, generado por la inteligencia artificial DeepMapsUsosSIOSE a partir de la imagen satelital de la zona delimitada por las siguientes coordenadas: 429327.3793037885, 454536.879329488, 4625543.069271753, 4651736.8192984555 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].

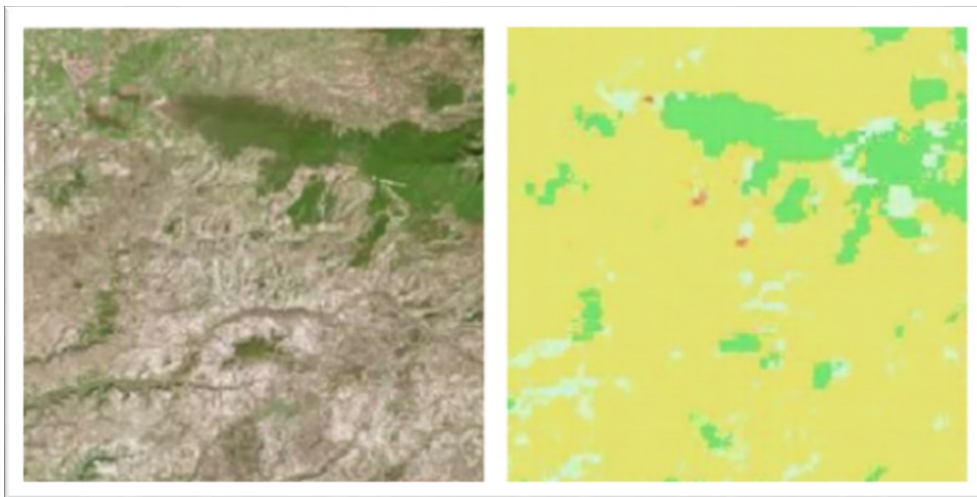


Figura 7.88. Ortoimagen generada por la inteligencia artificial DeepMapsUsosNatura a partir de ortoimagen satélite de Sentinel 2
Fuente: elaboración propia a partir de los datos del IGN

7.11.2. Caso 11: Ejemplo 2: Mapa de usos del suelo de zona sur de Burgos

En este ejemplo se expone un mapa de usos del suelo generado por la inteligencia artificial DeepMapsSuperficies a partir de una ortofotografía satelital del sur de Burgos, a la altura de Madrigalejo del Monte, concretamente del área comprendida en las coordenadas: 425193.26471622573, 462615.9314210631, 4650017.027630033, 4684942.027665656 [EPSG:25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.89. Ortoimagen generada por la inteligencia artificial DeepMapsUsosNatura a partir de ortοimagen satélite de Sentinel 2
Fuente: elaboración propia a partir de los datos del IGN

7.11.3. Caso 11: Ejemplo 3: Mapa de usos del suelo del área entre Cuéllar y Cantalejo

Este ejemplo muestra la generación automatizada de un mapa de usos del suelo a través de la inteligencia artificial DeepMapsSuperficies, partiendo de una ortofotografía satelital del área comprendida entre Cuellar y Cantalejo, correspondiente a las coordenadas 398625.12927248934, 423009.1292973473, 4566996.0692120725, 4592650.069238225 [EPSG:25830 (ETRS89 / UTM zona 30N - Proyectado)].



Figura 7.90. Ortoimagen generada por la inteligencia artificial DeepMapsUsosNatura a partir de ortoimagen satélite de Sentinel 2

Fuente: elaboración propia a partir de los datos del IGN

En los ejemplos mostrados en este caso puede comprobarse como la inteligencia artificial ha asignado correctamente las categorías de “Forestal”, “Agricultura” y “Zonas terrestres sin uso económico” a los diferentes elementos visibles en las ortofotografías del Sentinel 2.

7.12. Caso 12 Generación automatizada de mapas de usos del suelo con categorías de Natura2000

En este caso se muestran varios ejemplos de aplicación de la red neuronal DeepMapsUsosNatura, para demostrar la efectividad de esta. En los diferentes ejemplos se van a utilizar fuentes distintas de datos, comprobándose así cómo se pueden generar nuevos mapas de uso de superficies que se adapten a lo recogido en distintos años por distintos vuelos. Esto resulta importante, ya que demuestra la posibilidad de generar nuevos mapas de usos del suelo mucho más actualizados y cubriendo una superficie notablemente mayor que hasta ahora, ya que los datos de usos de superficies de la red Natura, ofrecidos por el IGN, no abarcan todo el territorio español y están muy limitados temporalmente (sólo existen datos de un año).

Dado que la red neuronal DeepMapsUsosNatura se ha entrenado para que sea capaz de generar mapas con las mismas categorías que los mapas de uso de superficies de la red Natura2000, la leyenda de los mapas generados por la red neuronal es la misma que la de la red Natura2000.

A continuación, se muestra la leyenda de la red Natura y por lo tanto la de los mapas generados por la red neuronal DeepMapsUsosNatura.

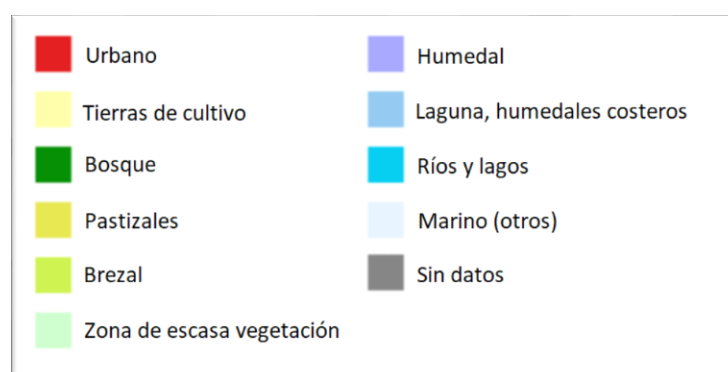


Figura 7.91. Leyenda de los mapas generados por la inteligencia artificial DeepMapsUsosNatura y de los datos utilizados para su entrenamiento
Fuente: elaboración propia a partir de los datos del IGN

7.12.1. Caso 12: Ejemplo 1: Creación de mapa de usos a partir de datos del SIGPAC

En este ejemplo se muestra un mapa de uso de superficies generado a partir de los datos del SIGPAC, como demostración de la eficacia del modelo DeepMapsUsosNatura sobre ortofotos distintas a las que se utilizaron para su entrenamiento (PNOA de Máxima Actualidad).

Los datos originales del SIGPAC, mostrados a continuación, pertenecen al archivo PNOA-H_SIGPAC_OF_ETRS89_HU30_h50_0564.ecw, descargado del IGN. La ortoimagen mostrada a continuación se corresponde con las coordenadas: 570960.5461340799, 571242.8036343668, 4487178.915036767, 4487440.852537033 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)]



Figura 7.92. Ortoimagen generada por la inteligencia artificial DeepMapsUsosNatura a partir de ortofotografía original del SIGPAC
Fuente: elaboración propia a partir de los datos del IGN

Tanto la imagen original del SIGPAC, como el mapa de usos de suelo generado a partir de la misma tienen una escala de 1:1.500. A continuación, se muestra como la inteligencia artificial DeepMapsUsosNatura es capaz de

identificar correctamente las categorías de la red Natura2000 de “bosques de coníferas”, “cursos de agua” y “pastizal o herbazal”.

7.12.2. Caso 12: Ejemplo 2: Generación automática de mapa de usos a partir de datos del PNOA de Máxima Actualidad

Este ejemplo muestra cómo la inteligencia artificial DeepMapsUsosNatura es capaz de actualizar los datos disponibles de la red Natura2000, en base a ortofotografías aéreas más actuales, concretamente los datos que ha procesado la inteligencia artificial en este ejemplo han sido extraídos de la hoja MTN50 0565 del PNOA de Máxima Actualidad. Para ilustrar este hecho se ha seleccionado un área limitada por las siguientes coordenadas: 569912.8044249644, 570190.3523419152, 4478857.682988552, 4479075.964238776 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)]. Todas las ortoimágenes mostradas en este ejemplo tienen una escala de 1:1.250.



Figura 7.93. Ortofotografía original del PNOA Máxima Actualidad
Fuente: elaboración propia a partir de los datos del IGN

En el área seleccionada puede verse el paso del Río Escabas a su paso por el sur de Fuertescusa (Cuenca). Cuando se elaboraron los datos de la capa Natura2000 utilizada para el entrenamiento, se incluyó este río; sin embargo, en los datos del PNOA más actuales, este tramo del río aparece seco; es por ello por lo que la inteligencia artificial no lo reconoce como un curso de agua. Además, ha sido capaz de actualizar los datos eliminando tanto un prado como un pastizal o herbal que en la actualidad han sido ocupados por un bosque de coníferas. Tal y como puede verse en las Figura 7.94 donde pueden compararse el mapa de usos de superficies de Natura2000 y el generado por la inteligencia artificial.

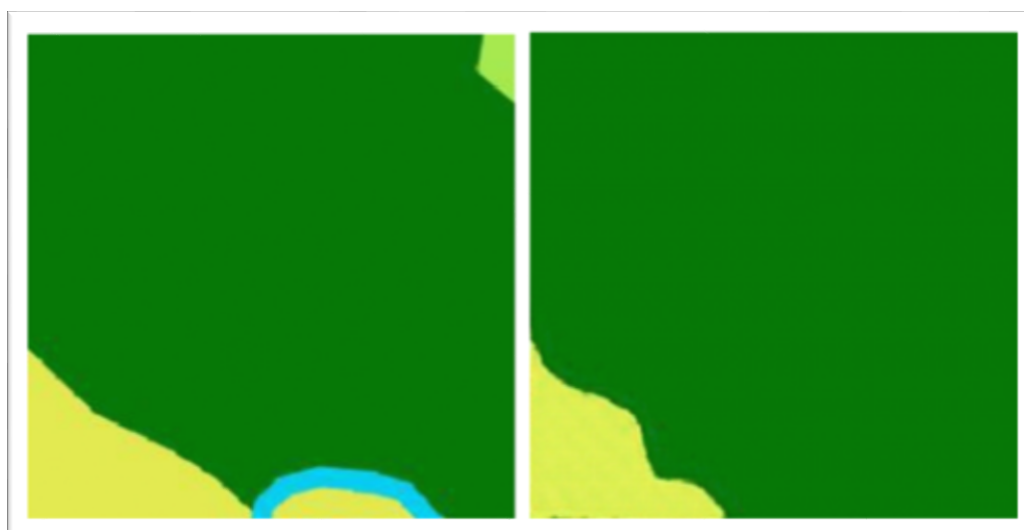


Figura 7.94. Comparativa entre ortoimagen original de Natura2000 y ortoimagen generada por el modelo DeepMapsUsosNatura

Fuente: elaboración propia a partir de los datos del IGN

Como puede apreciarse en la comparativa anterior, la inteligencia artificial ha sido capaz de representar más fielmente la realidad actual que la red natura, recurriendo a la categoría de bosques y a una categoría híbrida entre bosques y pastos que utiliza para representar la zona de pastos que contiene algunos árboles dispersos.

7.12.3. Caso 12: Ejemplo 3: Generación automática de mapa de usos a partir de datos del PNOA 2018

En este ejemplo se muestra la creación automatizada de un mapa de usos a partir de los datos del archivo PNOA_ANUAL_2018_OF_ETRS89_HU30_h50_0539.ecw del IGN, correspondiente a la hoja MTN50 0539 del PNOA 2018. Para ilustrar este ejemplo se ha seleccionado el área limitada por las siguientes coordenadas: 578642.3126584038, 578930.9624920322, 4490996.426462505, 4491223.438962737 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].

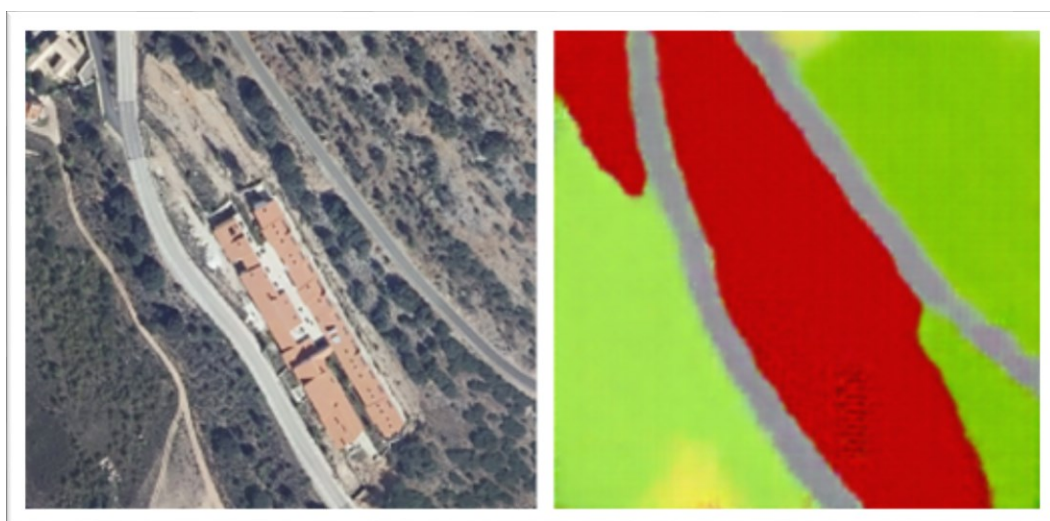


Figura 7.95. Ortoimagen generada por la inteligencia artificial DeepMapsUsosNatura a partir de ortofotografía original del PNOA 2018

Fuente: elaboración propia a partir de los datos del IGN

Se ha escogido esta área porque es un buen ejemplo de cómo la inteligencia artificial es capaz de identificar correctamente, no sólo la cubierta vegetal del terreno sino también las carreteras y las áreas urbanas.

Tanto la imagen original del PNOA como el mapa de usos generado con la inteligencia artificial DeepMapsUsosNatura tienen una escala de 1:1.300.

7.12.4. Caso 12: Ejemplo 4: Generación automática de mapa de usos del Monte de Valdelatas a partir de datos del PNOA de Máxima Actualidad

Este ejemplo es otra muestra de cómo la inteligencia artificial DeepMapsUsosNatura es capaz de generar mapas de uso de superficies en lugares que no han llegado a ser mapeados por la red Natura2000.

Otro ejemplo más extremo de esta cuestión es el siguiente mostrado en este caso, en el que se muestra como la red DeepMapsUsosNatura genera un mapa de usos del suelo de Tasmania, con las características de los mapas proporcionados por la red Natura2000.

En este ejemplo, el mapa de usos generado corresponde a una sección del Monte de Valdelatas, próxima al margen sur de este monte, concretamente la limitada por las siguientes coordenadas: 442822.22158096335, 443061.89174787444, 4486161.169447304, 4486392.430134674 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)].

El mapa de usos del suelo mostrado a continuación ha sido generado a partir de una ortoimagen del PNOA de Máxima Actualidad.

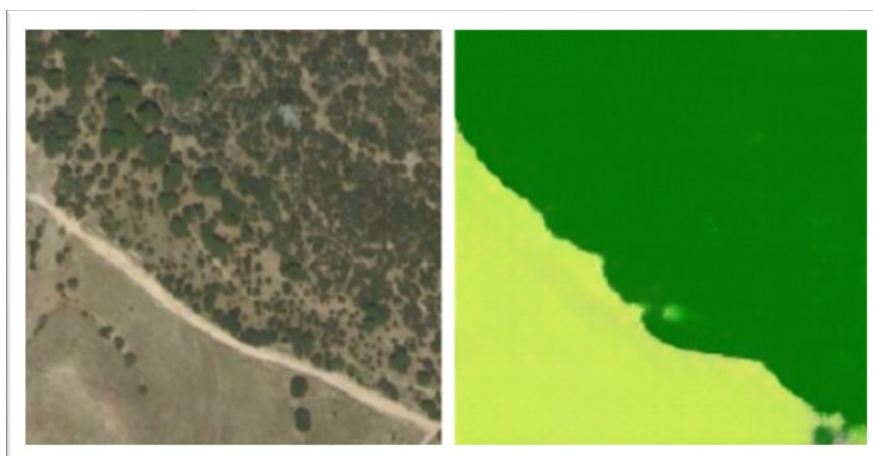


Figura 7.96. Mapa de usos del suelo generado por la inteligencia artificial DeepMapsUsosNatura a partir de una ortofotografía original del PNOA Máxima Actualidad
Fuente: elaboración propia a partir de los datos del IGN

Tanto la ortoimagen original del PNOA de Máxima Actualidad como el mapa de usos generado por la inteligencia artificial que se muestran a continuación, tienen una escala de 1:1.300.

En algunas zonas del Monte de Valdelatas, puede apreciarse un fenómeno interesante observado con las inteligencias artificiales entrenadas para generar mapas de uso de superficies o mapas de densidad de árboles. Las redes neuronales que han sido entrenadas para la creación de mapas temáticos, cuando son entrenadas lo suficiente, son capaces de generar, por sí solas, nuevas categorías híbridas entre las categorías existentes en los datos que se usaron para su entrenamiento y de esta forma representar mejor la realidad; este fenómeno ya aparece en otros de los ejemplos presentados en este capítulo, como en el ejemplo anteriormente expuesto de “El Pardo” (Madrid). En las siguientes imágenes puede comprobarse cómo el modelo DeepMapsUsosNatura ha utilizado combinaciones de las categorías existentes de pastizal, matorral y coníferas para representar de la forma más fiel a como lo habría hecho red Natura2000 para representar una realidad que no encajaría en las categorías preexistentes.

Las ortoimágenes mostradas a continuación se corresponden con las siguientes coordenadas: 441677.1596679528, 441908.85000152246, 4487099.59149207, 4487326.603992302 [EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado)]. La escala a la que están representadas es de 1:1.300.

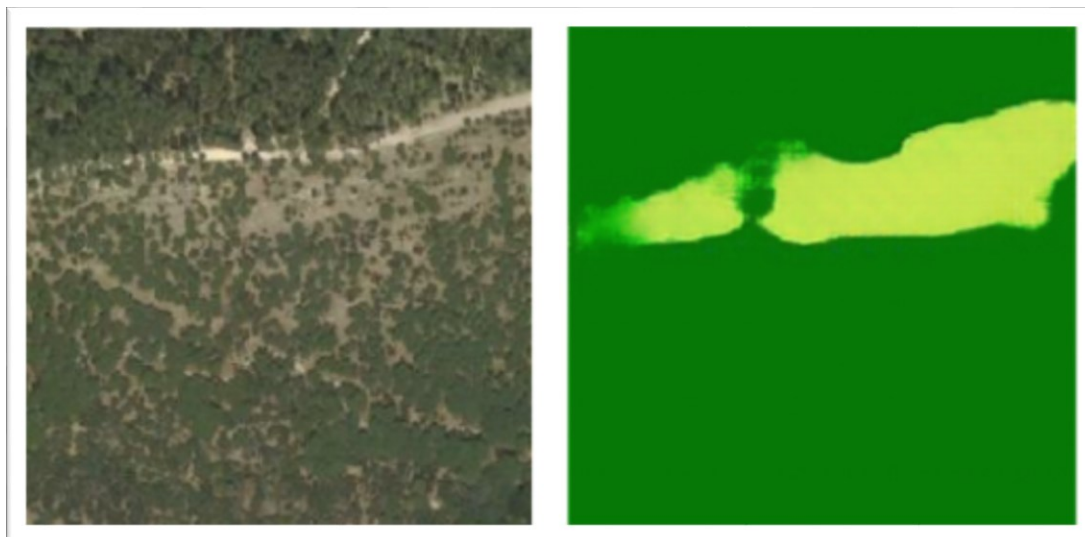


Figura 7.97. Imagen original del PNOA de Máxima Actualidad (izquierda) y mapa de usos, generado a partir de la misma por la inteligencia artificial DeepMapsUsosNatura (derecha), en el que se muestran las categorías híbridas y los gradientes producidos sin haber recibido entrenamiento para ello

Fuente: elaboración propia a partir de los datos del IGN

7.12.5. Caso 12: Ejemplo 5: Generación automática de mapa de usos a partir de datos de la capa Esri Aéreo del parque South Western National Park

Este ejemplo, al igual que algunos vistos anteriormente en este capítulo, muestra la capacidad de adaptación de las inteligencias artificiales creadas a zonas alejadas de aquellas de las cuales se obtuvieron los datos de entrenamiento. En este ejemplo, además, puede comprobarse cómo las redes neuronales son capaces de procesar correctamente datos de fuentes distintas sin necesidad de realizar adaptación de dominio, siempre y cuando los datos de entrenamiento y los datos sobre los que se van a aplicar tengan características similares.

Este ejemplo demuestra que las redes neuronales entrenadas con datos del PNOA trabajan bien con datos de ortoimágenes aéreas de Esri. En este

ejemplo, además, puede verse cómo las redes neuronales cartográficas son capaces de realizar su función correctamente, con independencia del sistema de referencia de coordenadas utilizado, ya que en esta ocasión para trabajar con los datos de Tasmania se ha utilizado el SRC con EPSG 102100, el cual es diferente al de los datos con los que la red ha sido entrenada.

Las ortoimágenes mostradas a continuación tienen una escala de 1:1.800 y el área que abarcan se corresponde con la zona del South Western National Park de Tasmania delimitada por las coordenadas: 16261815.29310269, 16262124.284103002, -5344460.816747129, -5344146.491746809 [EPSG: 102100].

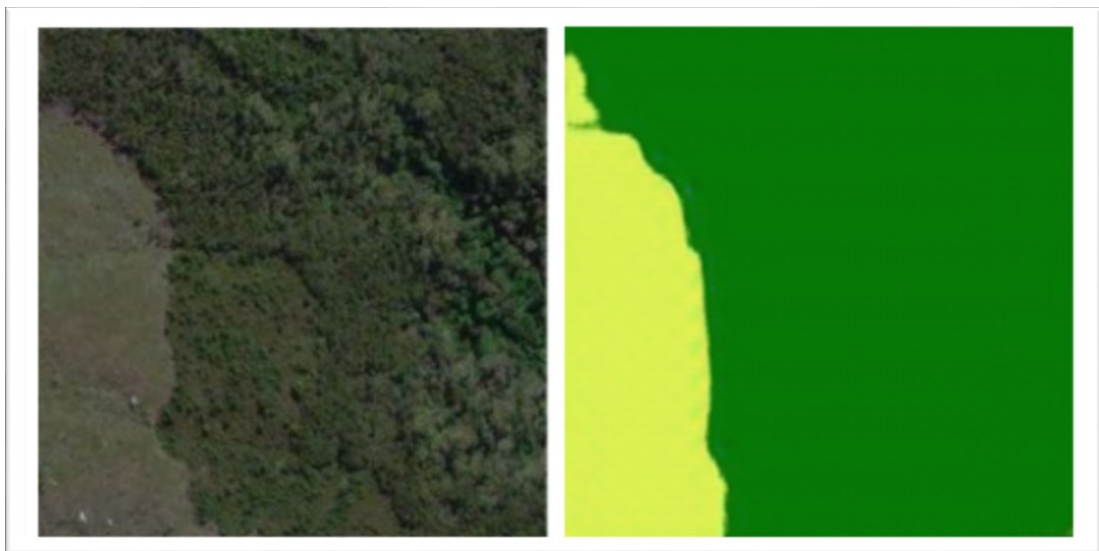


Figura 7.98. Mapa de usos del suelo generado por la inteligencia artificial DeepMapsUsosNatura a partir de una ortofotografía original de la capa Esri Aéreo
Fuente: elaboración propia, a partir de los datos de Esri

8. Elaboración del software 3D

En este capítulo se detallan los pormenores de la creación de las dos herramientas principales de procesamiento de datos tridimensionales desarrolladas durante esta tesis.

Para la elaboración de los distintos softwares de tratamiento y visualización de datos tridimensionales han debido tenerse en cuenta determinados conceptos, principios y técnicas propias del Big Data, los cuales, a su vez, determinan las metodologías de trabajo propuestas en el capítulo 9.

8.1. Calidad de los datos

Para la elaboración de los diferentes softwares se ha recurrido, no sólo a la inteligencia artificial, sino también a la aplicación de determinadas técnicas utilizadas en Big Data. Debido a esto, es imprescindible resaltar la importancia de la calidad de los datos que se utilizan, lo que se conoce como Data Quality.

El concepto de Data Quality engloba las técnicas, procesos y operaciones orientados a la mejora de la calidad de los datos existentes.

La calidad, como disciplina tecnológica, provee de una serie de procedimientos, estándares y buenas prácticas relacionadas con el buen uso y transformación de los datos, el objetivo final es poder medir y adaptar la calidad de los datos utilizados, garantizando así su fiabilidad y exactitud, de forma que aporten todo su valor.

El software de escritorio MEL ha sido diseñado para facilitar al usuario las operaciones necesarias para garantizar la calidad de los datos.

8.1.1. Etapas del proceso de Data Quality

8.1.1.1. Data Assessment - Profiling (Perfilado del Dato)

Es el proceso de examinar los datos que existen en las fuentes de datos y recopilar estadísticas e información sobre los mismos. Este proceso se facilita mediante el software MEL a través de la opción “Información de archivo .LAS (Genera HTML)”, incluida en el menú archivo.



Figura 8.1. Detalle de la herramienta, para la generación de informes a partir de los datos contenidos en los archivos LiDAR, incluida en el menú archivo del software MEL

Fuente: elaboración propia

Esta opción permite obtener todos los datos y estadísticas de un archivo LiDAR mediante la generación automática de un informe en HTML.

Entre otros datos, este informe incluye el número de puntos que contiene el archivo LiDAR, el software con el que fue generado (en el caso de los archivos procedentes del IGN es “TerraScan”), los rangos de intensidad de la señal, los máximos y mínimos en los 3 ejes espaciales (X,Y y Z), el datum, el sistema de referencia de coordenadas, además de las estadísticas generales en cuanto a la distribución de los puntos, a continuación se muestra un ejemplo de una de las tablas estadísticas generadas en cada uno de los informes (en concreto la distribución por clases).

Tabla de clasificación de puntos		
Clasificación	Número de puntos	Porcentaje
Unclassified	81249	2.1%
Ground	1134156	29.6%
Medium vegetation	664397	17.3%
Low point (noise)	5	0.0%
Reserved	1955136	51.0%

La tabla mostrada anteriormente ha sido extraída del informe generado automáticamente a partir de los datos del archivo de nube de puntos PNOA_2009_Lote4_CLM_554-4496_ORT-CLA-CIR.laz del IGN.

8.1.1.2. Transformación del dato

Conjunto de acciones que modifican la estructura, enriquecen o eliminan errores presentes en los datos.

Dentro de las transformaciones de datos posibles, cabe destacar el filtrado de los datos, que consiste en remover registros o campos enteros. En general, se utiliza durante la limpieza de datos, pero también se puede utilizar para cambiar la granularidad del dataset, tal y como ocurre cuando definimos la resolución en el software MEL.

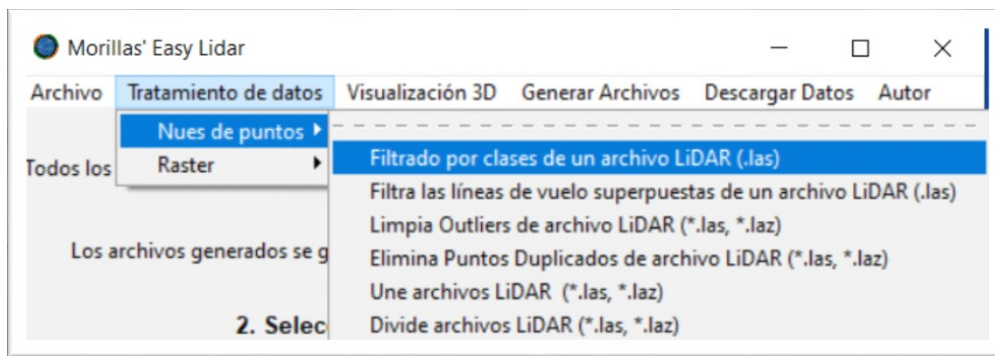


Figura 8.2. Detalle de la opción del menú de tratamiento de datos para el filtrado de las nubes de puntos de los archivos LiDAR
Fuente: elaboración propia

Uno de los procesos clave en el proceso de transformación o tratamiento del dato es el Data Cleansing (limpieza de datos), a través del cual se pueden detectar y luego corregir datos corruptos, incoherentes o erróneos de un dataset. Después del proceso, la información será consistente con otros conjuntos similares de datos. El objetivo no es borrar información sino mejorar la calidad del dato, construyendo un proceso de mejora continua. Este proceso permite detectar entradas duplicadas, incompletas, erróneas, etc. y establecer reglas para corregirlas

El proceso de limpieza de datos tiene como objetivo obtener datos limpios que permitan hacer análisis fiables y corregir los errores que haya habido en la adquisición de éstos.

El proceso de Data Cleansing se centra básicamente en 3 acciones sobre los datos:

Estandarizar: Transformar los datos a un formato común, en el caso del software MEL ese formato común sería el GeoTiff.

Parsear (Separar): Dividir un registro en partes según un patrón o unas reglas preestablecidas, como sucede en las consultas con el SIG de realidad

virtual o en la interpretación de los datos ingestados en el software MEL.

Corregir: Eliminar errores presentes en los datos; para llevar a cabo esta corrección de datos se han incluido numerosas funciones que se realizan de forma automática al lanzar los análisis automatizados incluidos en el software MEL, o de forma manual mediante los procesos incluidos en el menú “Tratamiento de datos” de este mismo software.

Dentro de la fase de limpieza de datos, suele aplicarse el Matching o Reglas de Comparación. Este proceso sirve para determinar cuáles de los registros han de considerarse duplicados; la técnica de matching puede utilizar reglas estrictas (que el dato sea exactamente igual) o mediante el uso de fuzzy logic (reglas de comparación difusa que internamente llevan algoritmos matemáticos).

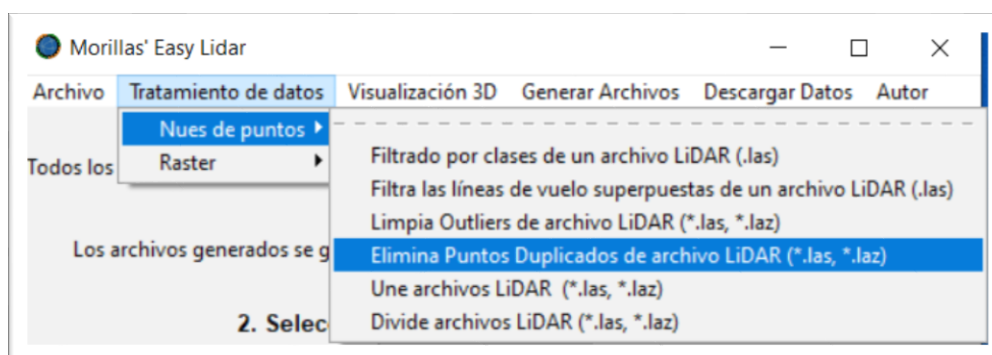


Figura 8.3. Detalle de la llamada al procedimiento de limpieza de datos duplicados a través del menú de tratamiento de datos del software MEL

Fuente: elaboración propia

En el software MEL, se ha incluido la posibilidad de realizar esta tarea mediante dos funciones incluidas dentro del menú de “Tratamiento de datos” : “Elimina Puntos duplicados” y “Filtra las líneas de vuelo superpuestas de un archivo LiDAR (.las)”, esta última función es especialmente importante para mejorar la calidad de los resultados obtenidos al trabajar con resoluciones altas, ya que los archivos de nubes de puntos suministrados por el IGN en ocasiones presentan pequeñas irregularidades en la medición de las alturas en las distintas

líneas de vuelo superpuestas en un mismo área, provocando así una merma en la calidad de los hillshades o los mapas hipsométricos si no se tratan adecuadamente antes.

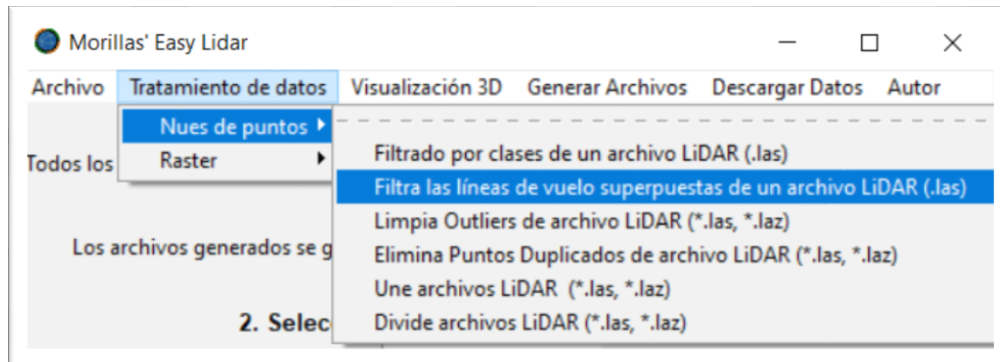


Figura 8.4. Detalle de la llamada al procedimiento de filtrado de las líneas de vuelo superpuestas a través del menú de tratamiento de datos del software MEL
Fuente: elaboración propia

8.1.1.3. Enriquecimiento de datos

Este concepto se refiere a las acciones que permiten incorporar nueva información al dataset. Puede consistir en insertar registros o campos adicionales desde otros datasets relacionados, o usar fórmulas para crear nuevos campos; como ejemplo de esto último tenemos los distintos procesos incluidos en el software MEL, los cuales posibilitan la creación de datos derivados, como son los modelos digitales del terreno, los mapas de pendientes, los hillshades, los mapas hipsométricos, los mapas de isolíneas o los mapas de acumulación de flujo. Estos procesos de generación de datos derivados son accesibles a través del menú “Generar Archivos”.

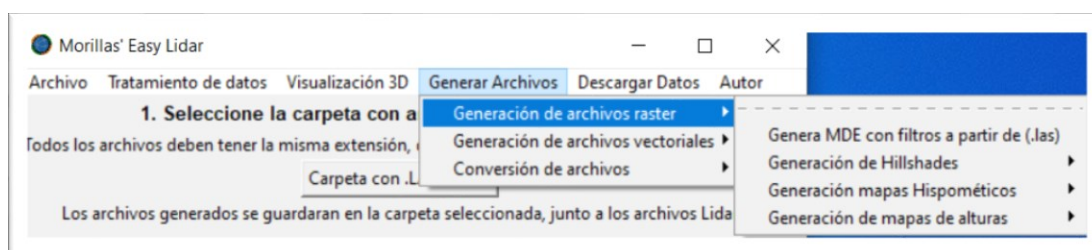


Figura 8.5. Ejemplo de algunas de las opciones presentes para la generación de datos derivados, incluidas en el menú “Generar Archivos”
Fuente: elaboración propia

8.1.1.4. Data Auditing o Data Validation

Por último, la auditoría de datos permite gestionar cómo los datos se ajustan a los propósitos buscados. Para facilitar este proceso y poder comprobar de forma masiva y automatizada las configuraciones paramétricas que permiten ajustar mejor los resultados obtenidos a los propósitos buscados, el software MEL incluye la generación automatizada de hillshades y mapas hipsométricos con distintas configuraciones de parámetros, para así poder escoger la configuración más adecuada para los fines buscados. Este tipo de procesos pueden aplicarse de forma masiva a todos los archivos contenidos en una carpeta, un ejemplo de parte de los resultados que se obtienen mediante estos procesos viene recogido en el anexo II.

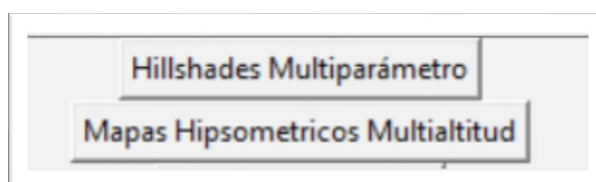


Figura 8.6. Botones presentes en la interfaz principal del software MEL, los cuales permiten lanzar el procesamiento masivo para la generación de archivos ráster con distintas configuraciones de parámetros
Fuente: elaboración propia

8.2. Desarrollo del software online

Durante el desarrollo de esta investigación se exploró la posibilidad de crear visores cartográficos con diversas tecnologías, para lo cual se realizaron una serie de pruebas, obteniendo distintos visores web basados en diferentes herramientas o librerías, parte de los cuales están disponibles en el portal (<http://www.cartotecnica.es/>).

La creación de entornos virtuales donde el usuario pueda moverse e interactuar con la información tiene una clara posibilidad de aplicación en el ámbito geográfico.

Al igual que podemos crear mundos virtuales a partir de nuestra imaginación, también podemos crear estos entornos virtuales a partir de datos geográficos.

Hace ya algunos años que los modelos tridimensionales se vienen utilizando dentro del ámbito geográfico, pero el reciente desarrollo del hardware, el abaratamiento de su coste y su portabilidad nos permite, en la actualidad, acceder a entornos virtuales desde nuestros dispositivos móviles, utilizando los mismos como gafas de realidad virtual, siendo únicamente necesario para ello un adaptador como Google Cardboard.

Desde hace ya algunas décadas se desarrollaron librerías que permitían el renderizado tridimensional de modelos; el desarrollo experimentado por estas tecnologías vino principalmente impulsado por la industria del entretenimiento, es decir los videojuegos. Como consecuencia de este impulso, hace ya algunas décadas, se creó la librería OpenGL, la cual utilizaba gran parte de los videojuegos para recrear escenarios tridimensionales. Con la mejora de las comunicaciones acontecida en los últimos años, así como la capacidad de los dispositivos, surgió la necesidad de adaptar las tecnologías de renderizado

tridimensional al mundo web; debido a ello aparecieron WebGL y librerías como three.js.

Gracias a estos desarrollos, en la actualidad es posible crear entornos web tridimensionales. A pesar de que el avance de estas tecnologías ha venido impulsado por la industria de los videojuegos, esta capacidad de transmisión de datos y de renderizado tridimensional puede ser aplicable en el ámbito de la geografía.

Por lo tanto, se plantea la posibilidad de crear un visor cartográfico web con capacidad de renderizado tridimensional y adaptación a la tecnología de realidad virtual basada en la generación de imágenes estereoscópicas para la visualización dinámica de grandes volúmenes de datos de muy diversas fuentes.

Para ello sería conveniente, en primer lugar, analizar el estado actual de las capacidades de renderizado tridimensional en el entorno web.

A principios del siglo XXI, se hicieron las primeras aproximaciones a la creación de entornos web tridimensionales, pero la integración con otras tecnologías y fuentes de datos aún era complicada.

Actualmente existen diversas formas de mostrar modelos tridimensionales mediante el uso de los navegadores web estándar.

En primer lugar, tendríamos que mencionar la posibilidad de renderizado 3D aportada por la última versión de CSS. El Cascading Style Sheets (CSS); se trata de un lenguaje web creado para diseñar el contenido HTML en una página web, modificando colores, tipos y tamaño de fuentes, sombras, imágenes, posicionamiento de elementos, etc. En sus últimas versiones se han incluido también las capacidades para crear efectos dinámicos, determinadas interacciones impulsadas por eventos activados por el usuario e, incluso, la

posibilidad de crear transformaciones de la perspectiva de forma tridimensional; gracias a esto podemos crear mediante este lenguaje representaciones tridimensionales accesibles mediante la web.

A pesar de la posibilidad de crear modelos tridimensionales con CSS, la integración de este lenguaje con los datos resulta complicada, ya que es un lenguaje creado para cuestiones de estilo principalmente. Debido a ello, se hace necesario siempre el uso de JavaScript para permitir una integración y manipulación efectiva de los datos.

JavaScript es un lenguaje de programación web, completamente integrado con los lenguajes CSS y HTML para el desarrollo web. Podríamos resumir estos conceptos de la siguiente manera:

- HTML: Lenguaje necesario para definir los elementos de las páginas web.
- CSS: Lenguaje necesario para definir el estilo de los elementos de las páginas web.
- JavaScript: Lenguaje necesario para programar el comportamiento de los elementos de las páginas web y el acceso a los datos.

En base a lo anteriormente expuesto, la forma más adecuada de crear un sistema de visualización cartográfica web es la combinación de los tres lenguajes, utilizando como base para las operaciones de carga de datos y de renderizado las librerías desarrolladas en JavaScript. A continuación, se muestra un ejemplo del resultado obtenido con el SIG online de Realidad Virtual.

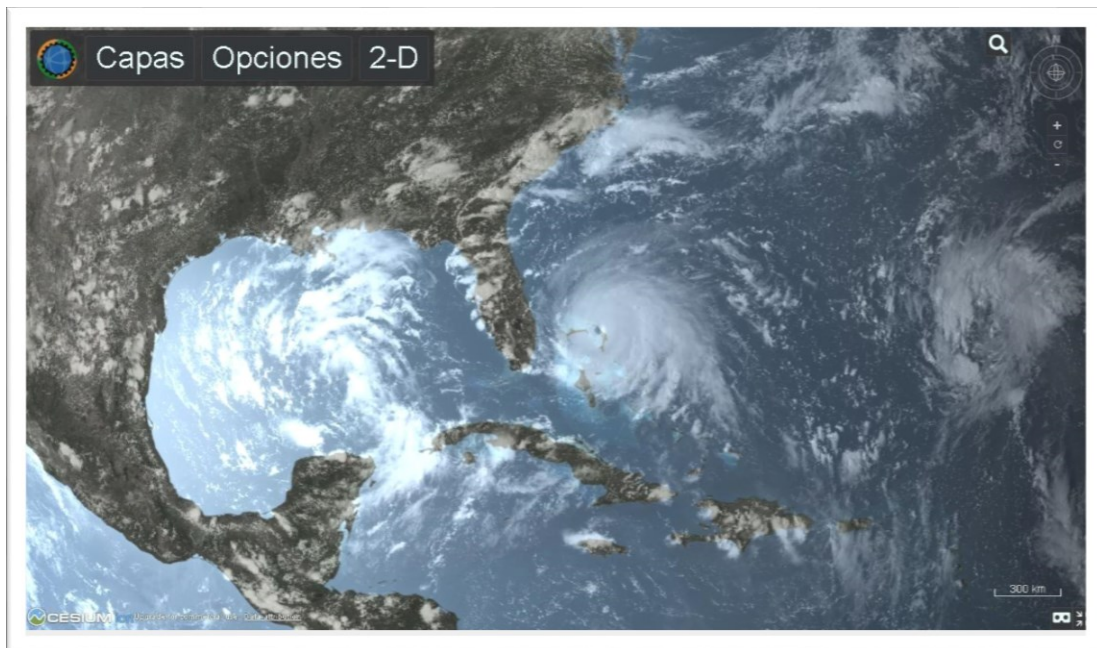


Figura 8.7. Ejemplo de imágenes atmosféricas a tiempo real, sobre el modelo tridimensional de la Tierra (huracán Dorian acercándose a Florida)
Fuente: elaboración propia, a partir de datos de Microsoft y la Nasa

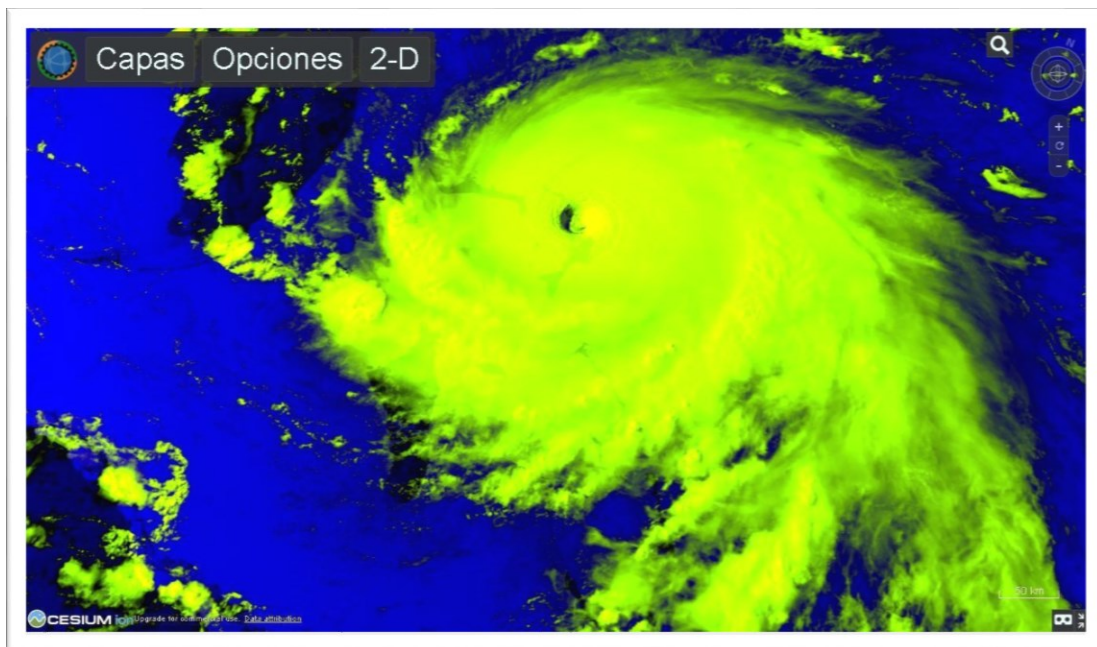


Figura 8.8. Espectro visible editado a tiempo real con el software online
Fuente: elaboración propia, a partir de datos de Microsoft y la Nasa

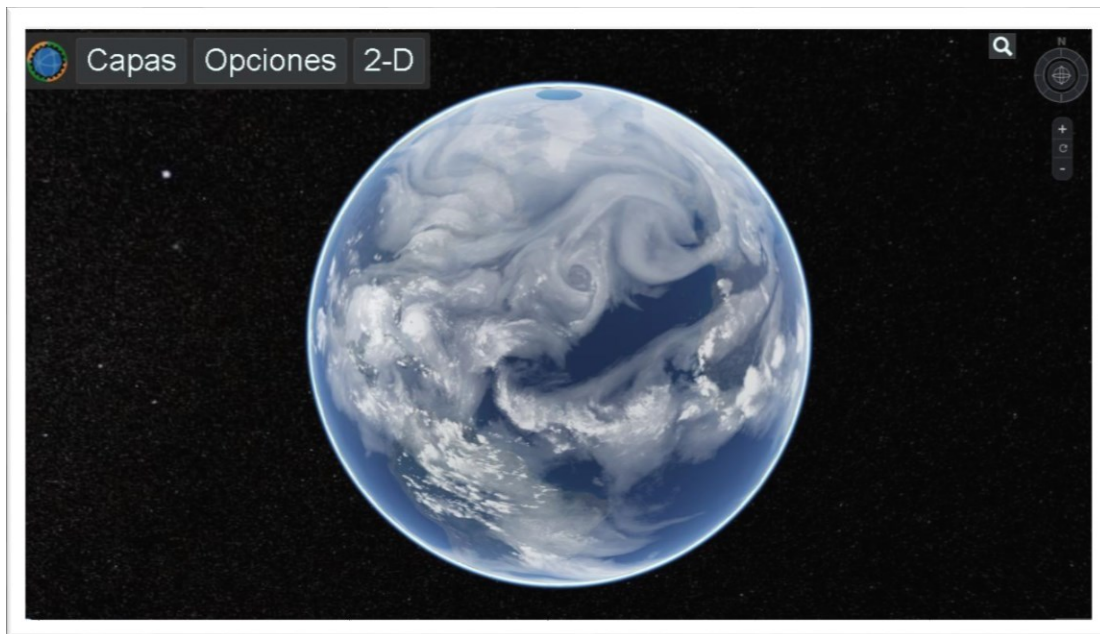


Figura 8.9. Capa de vapor de agua superpuesta al geoide
Fuente: elaboración propia, a partir de datos de Microsoft y la Nasa

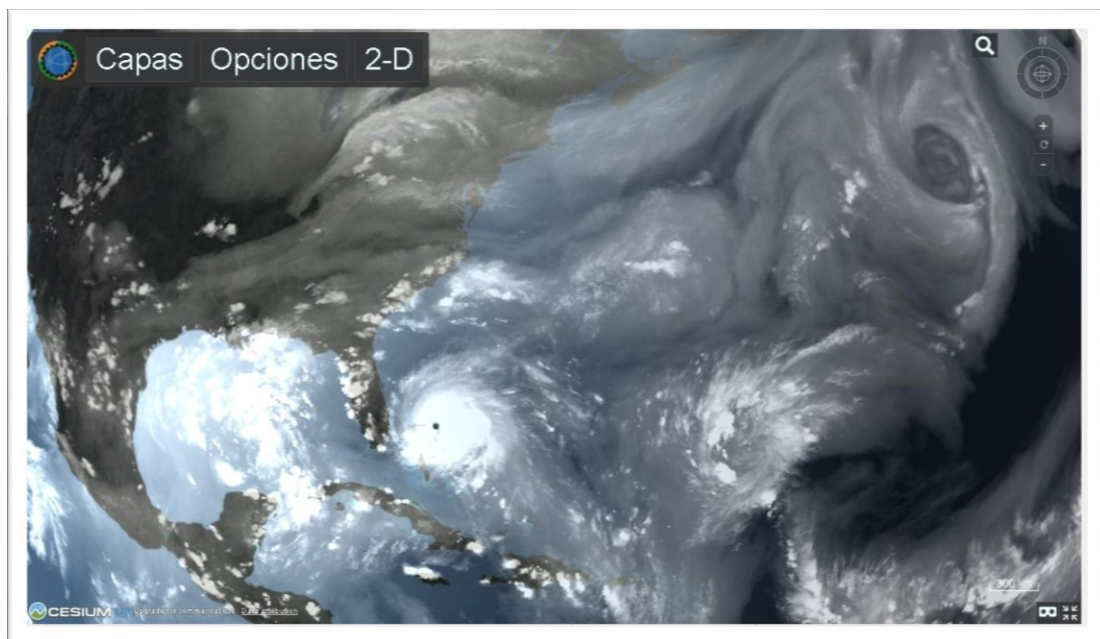


Figura 8.10. Detalle de la capa de vapor de agua de la Nasa
Fuente: elaboración propia, a partir de datos de Microsoft y la Nasa

En las imágenes mostradas anteriormente puede verse el huracán Dorian a las 23:00 del día 1 de septiembre de 2019, acercándose a Florida.

La librería Cesium, utilizada para el desarrollo del software online de Realidad Virtual, permite, además de la renderización de modelos tridimensionales, la obtención de imágenes estereoscópicas de los mismos, por lo que el visor desarrollado puede adaptarse para su uso mediante gafas de realidad virtual. A continuación, se muestra una imagen del renderizado estereoscópico del modelo.



Figura 8.11. Ejemplo de renderizado para Realidad Virtual, mediante la generación de imágenes estereoscópicas

Fuente: elaboración propia, a partir de datos de Microsoft

Este es el motivo por el que se ha utilizado esta librería para todos los visores que incluyen Realidad Virtual.

El menú que nos permite acceder a los distintos visores con Realidad Virtual elaborados durante la investigación es accesible a través de la siguiente URL: <http://www.cartotecnia.es/menus/vr/>. O a través del menú principal de [cartotecnia.es](http://www.cartotecnia.es), haciendo clic en el elemento “GIS con Realidad Virtual”.

Para una mayor inmersión en la simulación, se han incluido, además, la lectura de los sensores de campo magnético y acelerómetros del móvil, para que cuando se acceda al visor a través de un dispositivo móvil el modelo digital responda a la posición y ángulo de nuestra cabeza.

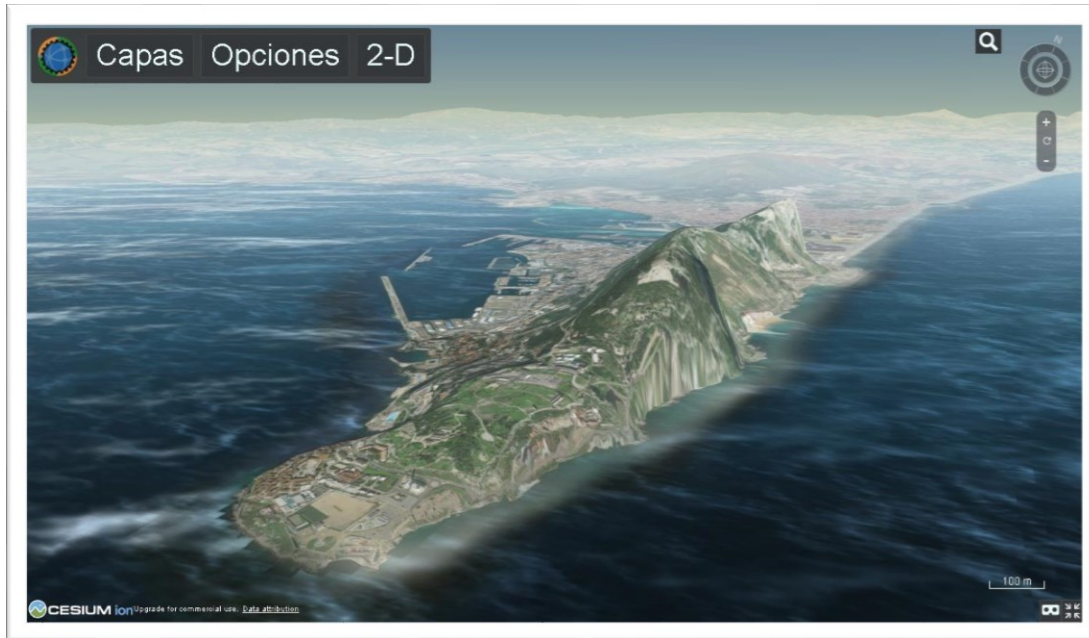


Figura 8.12. Imagen del Peñón de Gibraltar, generada con el visor online
Fuente: elaboración propia, a partir de datos de Microsoft, la Nasa, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia

Tal y como se muestra en la Figura 8.8, el SIG online de Realidad Virtual permite realizar el tratamiento de las imágenes que componen las capas a tiempo real mientras se utiliza el visor.

Las capas que se han dispuesto en los visores web de Realidad Virtual se dividen en capas base y capas superpuestas.

8.2.1. Lenguajes y librerías

Para el desarrollo de este software, se han utilizado exclusivamente lenguajes y librerías web. La mayoría de las funciones están implementadas en JavaScript, mientras que el estilo se ha desarrollado principalmente en CSS, utilizando HTML para crear la estructura de la página e integrar todos los elementos y librerías. Para la carga de capas y su renderizado se ha utilizado la librería Cesium.js, el estilo se ha diseñado utilizando la librería Bootstrap 3.3.7 y, para agilizar el desarrollo, la librería JQuery 3.2.1.

8.3. Desarrollo del Software de escritorio “MEL”

Para poder realizar un análisis del terreno, aprovechando la máxima resolución disponible del relieve de España (los datos contenidos en las nubes de puntos suministradas por el centro de descargas del IGN en formato .laz), se ha desarrollado el software Morillas Easy LiDAR, con el cual puede obtenerse una resolución máxima de 10 centímetros por pixel en los ficheros con mayor densidad de puntos por metro cuadrado. La alta resolución obtenida por el software desarrollado es posible gracias a la generación de superficies TIN (red irregular de triángulos), utilizando el algoritmo de triangulación de Delaunay. Este algoritmo intenta crear una superficie formada por triángulos, uniendo los puntos más próximos entre sí, para lo cual se crean círculos, tomando como centro cada uno de los puntos, y sus intersecciones se conectan a una red de triángulos no superpuestos y tan compactos como sea posible. A pesar de que el software desarrollado se ha centrado en proveer al usuario de herramientas que faciliten el tratamiento de los datos suministrados por el IGN, es capaz también de procesar correctamente nubes de puntos y modelos digitales del terreno procedentes de otras fuentes, un ejemplo de ello es la capacidad del software MEL para analizar y tratar los datos del Servicio Geológico de Estados Unidos (USGS). La siguiente imagen ha sido obtenida mediante la unión y posterior renderización tridimensional con coloración por elevaciones de las nubes de puntos USGS_LPC_PR_PuertoRico_2015_20QKF15505100_LAS_2018.laz y USGS_LPC_PR_PuertoRico_2015_20QKF15505250_LAS_2018.laz del USGS, todo ello realizado con el software MEL.

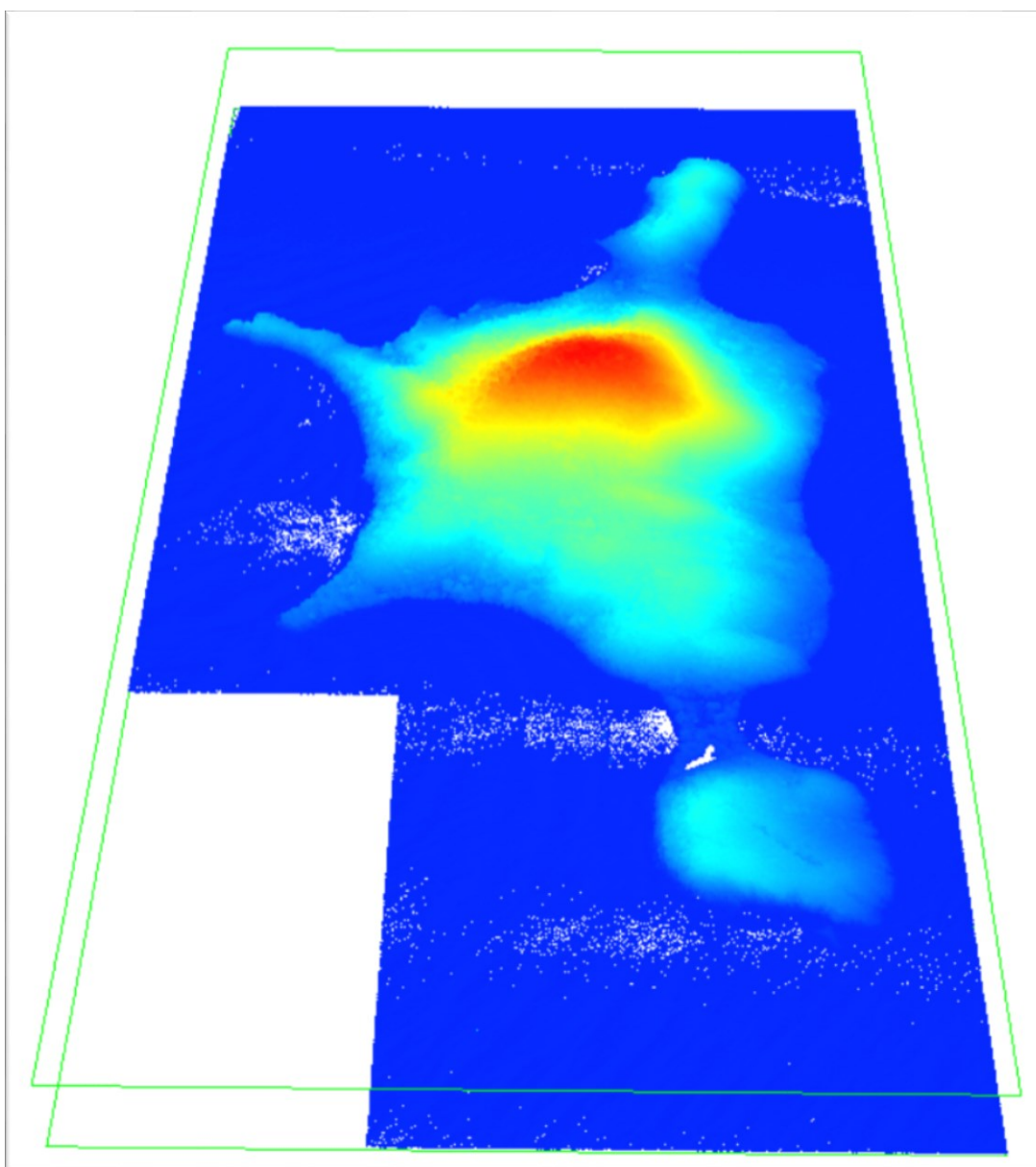


Figura 8.13. Visualización tridimensional de las nubes de puntos correspondientes al Cayo Luis Peña (pequeña isla frente a la costa oeste del municipio de Culebra, Puerto Rico)
Fuente: elaboración propia a partir de datos del USGS

La interpolación TIN (Red irregular de triángulos) ha sido la escogida para generar los ráster derivados de las nubes de puntos mostrados en esta tesis, debido a su idoneidad a la hora de representar relieves con detalle, pero el software MEL ofrece la posibilidad de trabajar con otros tipos de interpolaciones, en concreto con las interpolaciones de distancia inversa ponderada (IDW) y de vecino más cercano (Nearest Neighbor).

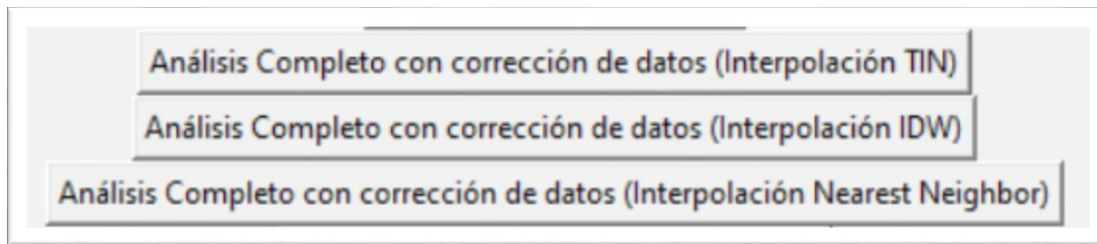


Figura 8.14. Detalle de los análisis basados en distintas interpolaciones disponibles en el software MEL

Fuente: elaboración propia

El orden en el que se muestran en la interfaz los distintos tipos de interpolación viene determinado por los resultados obtenidos mediante su aplicación.

La interpolación TIN, utilizada en el software que se ha desarrollado, se basa en la triangulación de Delaunay, la cual trata de crear una superficie formada por triángulos, tomando como vértices los puntos vecinos más cercanos. Para ello, se crean circunferencias alrededor de los puntos de muestra y sus intersecciones se conectan a una red de triángulos no superpuestos y lo más compactos posible.

La interpolación de distancia inversa ponderada se basa en la asignación de pesos a los datos incluidos en el entorno a analizar, definido por el radio; en nuestro caso el radio establecido para el cálculo de pesos ha sido de 5 metros. Mediante los datos presentes dentro de este radio se extrapola el valor del punto situado en el centro, considerando para la asignación de los pesos que la influencia de los datos en la asignación de pesos es inversamente proporcional a la distancia que los separa del punto a generar. En el caso de los algoritmos utilizados para la interpolación en software que ha sido desarrollado en esta investigación, se ha escogido un exponente de ponderación de 1, lo cual determina la contribución de los puntos circundantes al punto a generar; al respecto, cabe mencionar que a medida que se aumenta el valor de este exponente de ponderación, lo hace también la contribución de los puntos más próximos.

Por otra parte, la interpolación del vecino más cercano o interpolación proximal es el método más simple de los 3 utilizados en el software implementado que, si bien no ofrece tan buenos resultados, resulta más ligero computacionalmente y por lo tanto más rápido. En esta interpolación se asigna al punto desconocido el valor del punto más próximo a él, en este caso se ha configurado el radio de búsqueda en 5 metros.

Dado que el propósito de este software es facilitar el trabajo con datos tridimensionales para la mayor parte de los usuarios, determinados parámetros utilizados en los algoritmos contenidos en este software se han definido mediante código con los valores que dan mejores resultados en la mayor parte de los casos, permitiendo que el usuario modifique sólo aquellos parámetros que pueden resultar más útiles para adaptar los resultados al propósito buscado, los cuales vienen prefijados en los valores más utilizados habitualmente. Con la misma intención de simplificación del trabajo cartográfico con datos LiDAR y modelos digitales de elevaciones, en la interfaz principal del programa se incluyen scripts para automatizar las tareas más habituales con nubes de puntos; sin embargo, el software incluye la posibilidad de hacer operaciones más concretas, accesibles mediante los menús situados en la barra superior.



Figura 8.15. Detalle de la barra de menús del software MEL
Fuente: elaboración propia

Aunque el software desarrollado ofrece la posibilidad de seleccionar entre los tres tipos de interpolaciones expuestos anteriormente, se ha establecido como interpolación, para todas aquellas operaciones en las que no se especifique el tipo de interpolación, la interpolación de red irregular de triángulos (TIN).

Para poder comprobar mejor qué configuración facilita más la apreciación del relieve, se han generado hillshades del MDT a partir de los MDE obtenidos con distintas combinaciones de algoritmos de interpolación y resoluciones de malla. Finalmente, se ha optado por establecer un tamaño de celda de 25 cm.

Una renderización previa de las nubes de puntos nos permite apreciar que existen una serie de outliers o valores atípicos (ej.: nubes, pájaros, ruido en el sensor, etc.) que complican la adquisición de conocimiento por parte del usuario. Debido a esto, se hace necesario incluir, en la metodología propuesta, unos procesos de tratamiento y limpieza de datos, previos a la generación de los MDT.

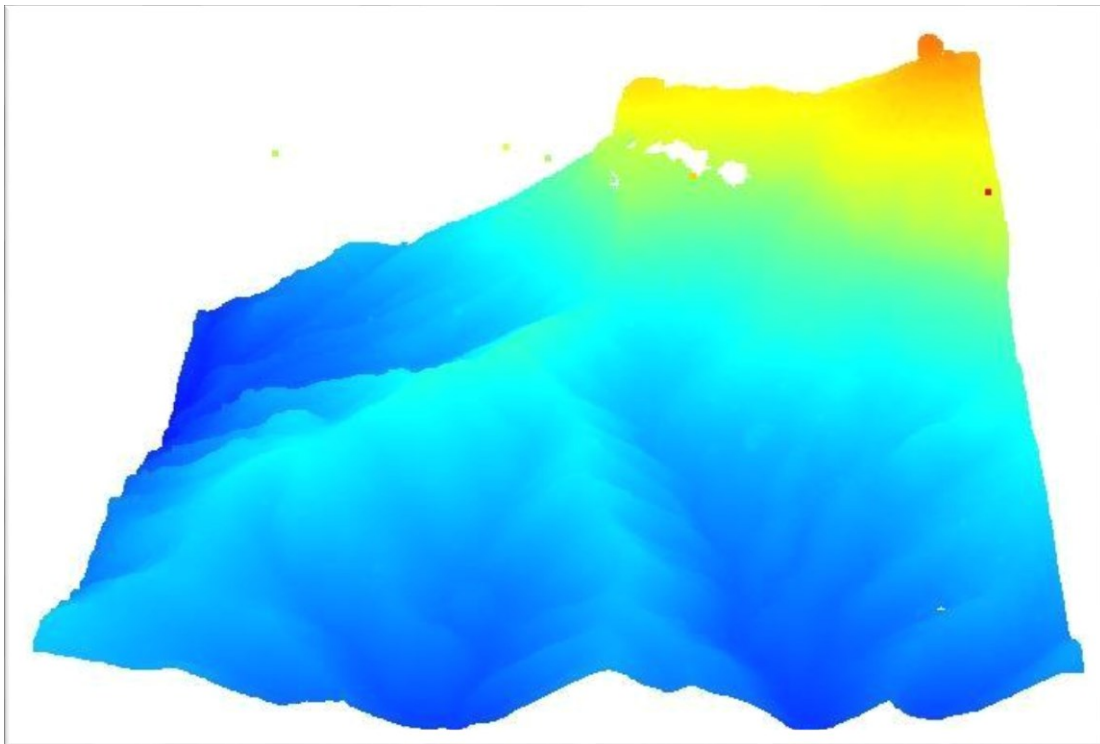


Figura 8.16. Ejemplo de datos erróneos en archivo de nube de puntos PNOA_2015_CANAR-GC_438-3096_ORT-CLA-RGB.laz (existen puntos aislados situados en el aire)

Fuente: elaboración propia a partir de datos del IGN

El proceso de limpieza de los puntos de la nube, que constituyen los valores atípicos, se basa en la generación de un TIN basado en los puntos clasificados como terreno y, a partir de aquí, calcular la altura de todos los demás puntos con ésta. Hecho esto, se eliminan todos los puntos que aparezcan a 40

metros o más sobre el nivel del suelo o 3 metros o más bajo el nivel del suelo, del archivo de salida. Este proceso está incluido en los scripts que se lanzan mediante los botones de la interfaz principal del software, pero también puede aplicarse de forma independiente a través del menú de “Tratamiento de datos”.

En los scripts de los distintos procesos automatizados, no sólo se eliminan los outliers o valores atípicos, sino que también se pueden limpiar aquellos valores que se encuentran repetidos o superpuestos. A esto hay que añadir la posibilidad de eliminar los objetos presentes en el terreno, mediante un análisis de la pendiente, estableciendo el criterio en una pendiente superior a 15 grados. Asimismo, se eliminan posibles grietas derivadas de la generación del modelo tridimensional, estableciendo la profundidad máxima de estas en 0.2 metros; de la misma forma se eliminan huecos o agujeros producidos por la ausencia de datos en un punto.

Debido a la falta de uniformidad existente en la densidad de puntos de las nubes de puntos ofrecidas por el IGN, además de incluir varios procedimientos analíticos para la corrección de los datos, eliminando los registros erróneos que generarían aberraciones en el modelado por interpolación TIN, y se han incluido otros que permiten el enriquecimiento de los datos, completando las zonas con menor densidad de puntos mediante la aplicación del algoritmo “k-nearest neighbors”.

Aunque los parámetros mencionados han sido los incluidos en los procesos automatizados, debido a que son los que mejor se ajustan a la mayor parte de la casuística, para casos particulares en los que estos procesos o la parametrización establecida por defecto pudiese no ser la adecuada, se ha incluido la posibilidad de que el usuario personalice la aplicación de los distintos procesos de limpieza y corrección de datos mediante las funciones incluidas en el menú “Tratamiento de datos” presente en la barra superior de la interfaz principal del software.

Como el uso principal para el que se ha desarrollado este software es el del estudio del relieve y la teledetección de estructuras de origen antrópico, se ha incluido en el proceso de generación de los MDT, a partir de la nube de puntos, un filtrado de los datos. Este proceso resulta fundamental, ya que muchas estructuras se encuentran cubiertas por la vegetación, siendo la tecnología LiDAR idónea para este propósito, puesto que, en función del reflejo del pulso láser, cada punto se clasifica, de forma automática, en varias categorías, las cuales incluyen la cubierta forestal, el agua o el terreno desnudo, entre otras; estas categorías vienen codificadas en el archivo .laz, mediante códigos basados en números enteros. Esta clasificación de los puntos permite la eliminación de los puntos correspondientes a elementos que dificultan la identificación de morfologías, como la vegetación alta, media o baja, los cuales dificultan la identificación de estructuras por los medios fotográficos o los modelos generados por fotogrametría convencionales, independientemente de la resolución de éstos.

Cabe mencionar que la eficacia de los filtros basados en la clasificación de los puntos, tanto los incluidos en los procesos automatizados como los que permiten la configuración manual por parte del usuario, depende de la buena categorización de los datos, lo cual no siempre se da en los archivos LiDAR del IGN, donde no es infrecuente encontrar errores como la asignación de la categoría de terreno a puntos correspondientes a cubierta vegetal; debido a ello, existen archivos en los que los datos procedentes de una de las líneas de vuelo que componen el archivo se encuentran mal categorizados, por lo que se requiere un procesado más exhaustivo y manual de los mismos, no pudiendo limitarse éste a la eliminación de determinadas categorías, debiendo incluir procesos de reclasificación de los puntos.

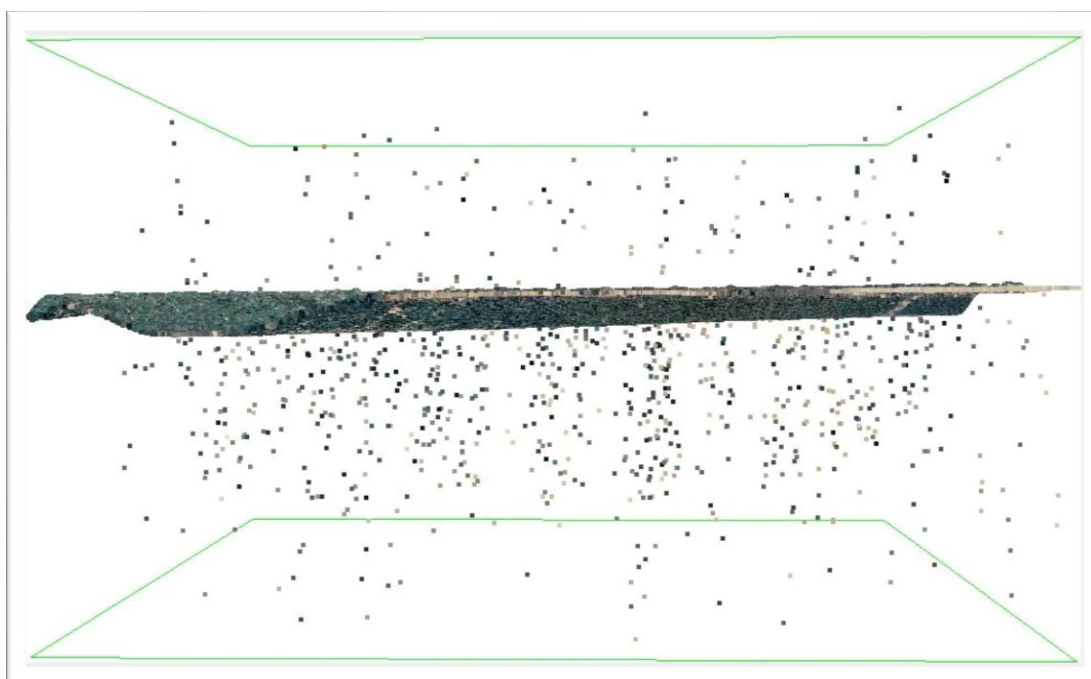


Figura 8.17. Datos erróneos, tanto sobre el terreno como bajo el mismo. Archivo LiDAR: PNOA_2009_Lote7_CLM_354-4370_ORT-CLA-COL.laz
Fuente: elaboración propia, a partir de datos del IGN

A partir del modelo generado, se obtiene el mapa de pendientes, que calcula la tasa máxima de cambio del valor de una celda a sus ocho celdas de vecindad, identificando la mayor inclinación hacia abajo de esa celda.

Por otra parte, para generar el GeoTiff de acumulación de flujo, se ha utilizado el algoritmo D-Infinity, propuesto por Tarboton en 1997; este algoritmo permite calcular el área contribuyente con mayor precisión en las laderas divergentes. Si bien no proporciona una solución completamente rigurosa a este problema de "tubo de flujo" o "tubo de corriente", es un enfoque bastante robusto y pragmático, que ofrece resultados visualmente atractivos. Es un algoritmo de dirección de flujo múltiple (MFD), que permite tanto ángulos de flujo continuo, como la partición de flujo entre uno o dos píxeles vecinos. Además, el algoritmo D-Infinity puede manejar todas las situaciones ambiguas que pueden ocurrir en la topografía real, mientras que muchos otros métodos MFD no pueden.

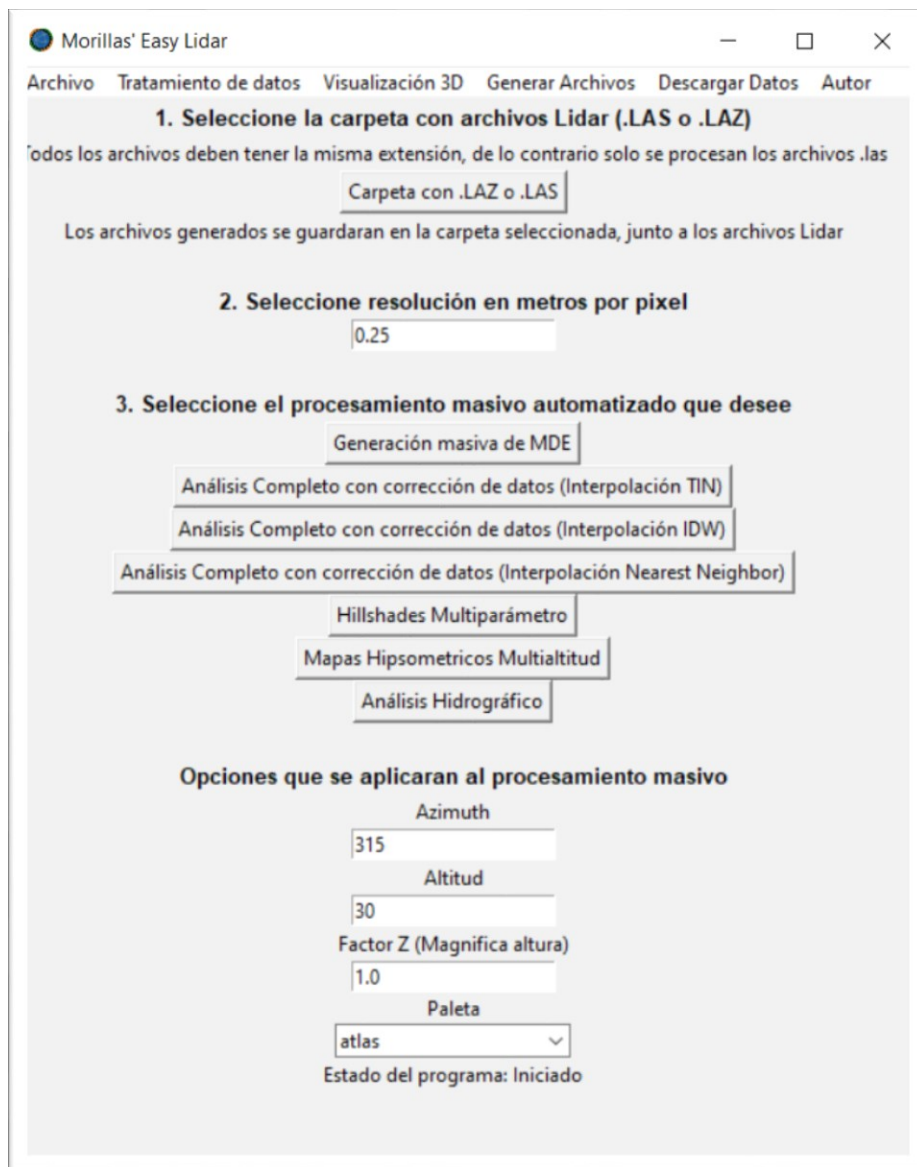


Figura 8.18. Interfaz gráfica del software MEL
Fuente: elaboración propia

A pesar de que, debido a la densidad de puntos existente en la mayor parte de los ficheros suministrados por el IGN, el algoritmo ofrece resultados visualmente más precisos para resoluciones de 0,25 píxeles por metro, razón por la cual se ha establecido esta resolución como la predefinida, el software permite cambiarla en cualquier momento mediante la modificación del valor numérico de coma flotante del cuadro de texto correspondiente al segundo paso a realizar por el usuario.

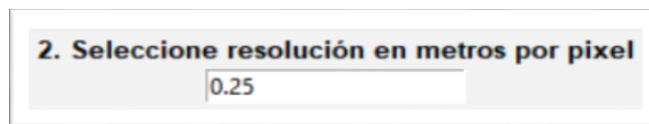


Figura 8.19. Detalle del cuadro de texto para establecer la resolución de los archivos ráster a generar
Fuente: elaboración propia

En el desarrollo de este software se ha optado por una GUI (Interfaz Gráfica de Usuario) simple, que facilite su uso y oriente al usuario en los pasos a seguir.

Debido a que una de las utilidades principales para las que se ha diseñado este software es la generación de MDE y la posterior generación de hillshades a partir de éstos, se han incluido también cuadros de texto, a través de los cuales el usuario puede modificar los valores de determinados parámetros (Azimuth, Altitud y Factor-Z) del hillshade a generar y la paleta de colores aplicable al mapa hipsométrico (la altitud del mapa hipsométrico se establece mediante el cuadro de texto “Altitud”, que también se utiliza para configurar el hillshade).

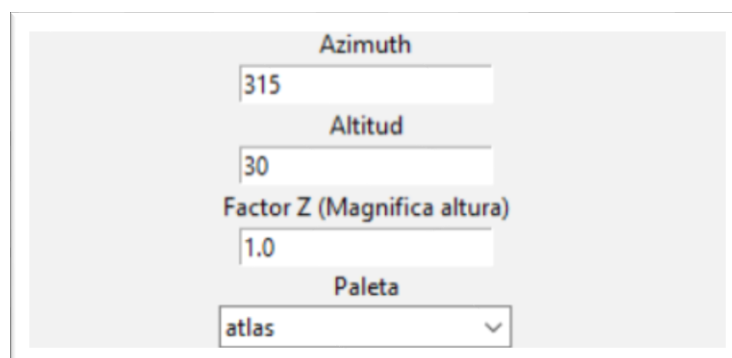


Figura 8.20. Detalle de cuadros de texto para ajustar parámetros de creación del hillshade y del mapa hipsométrico
Fuente: elaboración propia

Para facilitar la labor de búsqueda de diferentes morfologías, se ha incluido en el software la generación automática de hillshades y mapas hipsométricos con distintas combinaciones de los parámetros mencionados, a través de las opciones “Hillshades Multiparámetro” y “Mapas Hipsométricos Multialtitud”. En el anexo II pueden verse ejemplos de la aplicación de estas funciones y las

ortoimágenes generadas de forma automática con distintas configuraciones paramétricas.

El software desarrollado también permite la generación más personalizada de archivos, pudiendo cambiar de esta forma algunos de los parámetros que vienen establecidos mediante código en los procesos de generación automatizada; éste es el caso de la generación de mapas hipsométricos, en los cuales, además de los parámetros de altitud y paleta configurables mediante la interfaz principal, pueden modificarse otros muchos, tal y como se muestra en la siguiente imagen.

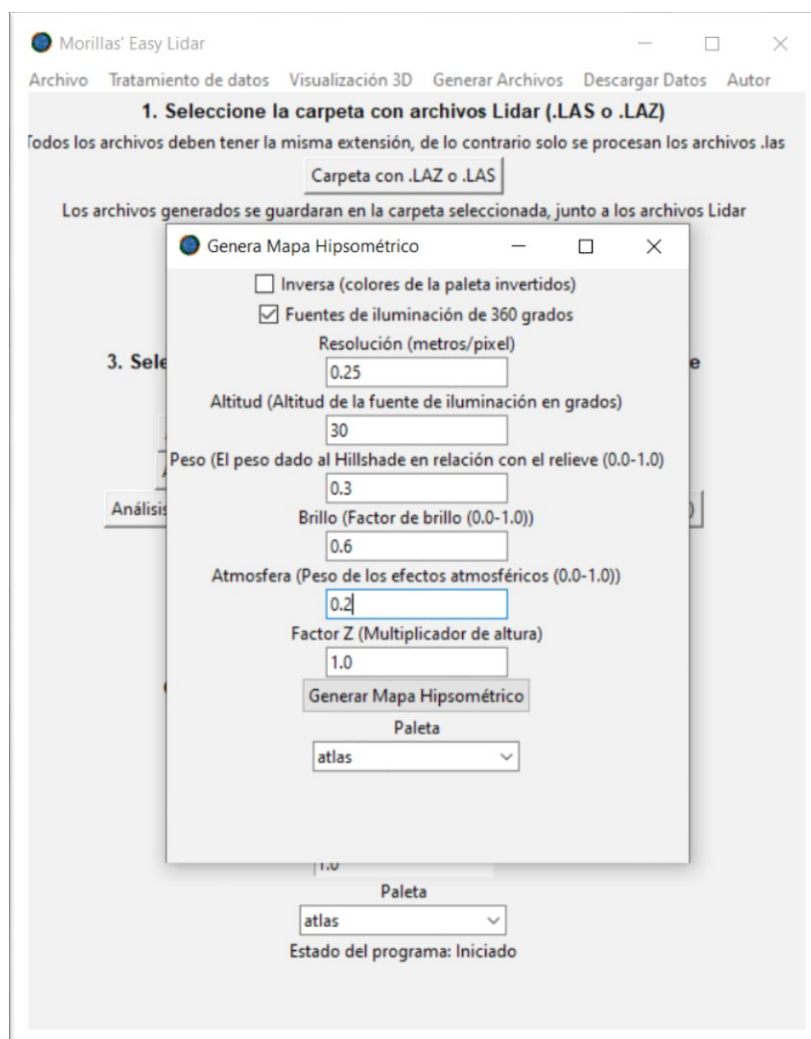


Figura 8.21. Interfaz secundaria para configuración manual de los parámetros de generación de mapas hipsométricos
Fuente: elaboración propia

Los parámetros con los que está configurada la generación automatizada de mapas hipsométricos, tanto multialtitud como los generados con las opciones de “Análisis Completo”, son los siguientes:

Altitud	30°
Peso	0.7
Brillo	0.6
Atmósfera	0.0
Paleta	Atlas
La fuente de iluminación está configurada para incidir desde todos los ángulos (360°)	

Por otra parte, la configuración predeterminada en la generación automatizada de hillshades es de 315° de azimuth, 30° de altitud y un factor z de 1.

Para establecer estos parámetros mediante el menú que se muestra en la Figura. 8.21, es necesario acceder mediante el menú “Generar Archivos”, tal y como se muestra en la siguiente imagen.

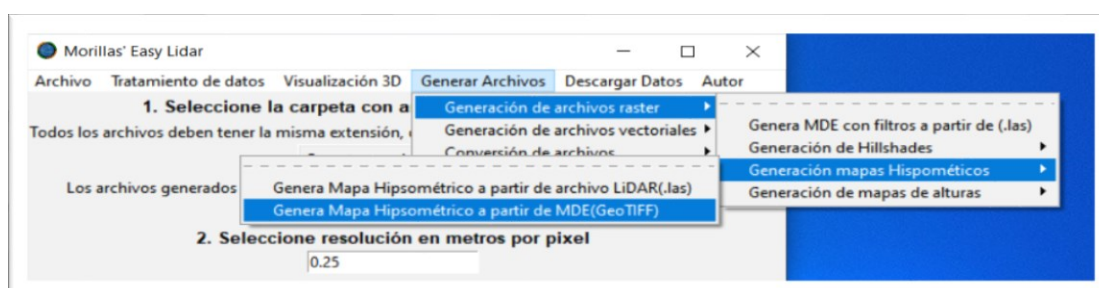


Figura 8.22. Opción del menú de generación de archivos que permite la creación de mapas hipsométricos parametrizados manualmente a partir de un modelo digital de elevaciones

Fuente: elaboración propia

Para permitir una mayor usabilidad del software, y para facilitar el trabajo con las imágenes generadas, se ha incluido también un visor/editor de imágenes ligero de software libre (irfanview64), que permite la aplicación de diferentes filtros y la variación de los parámetros de la imagen generada.

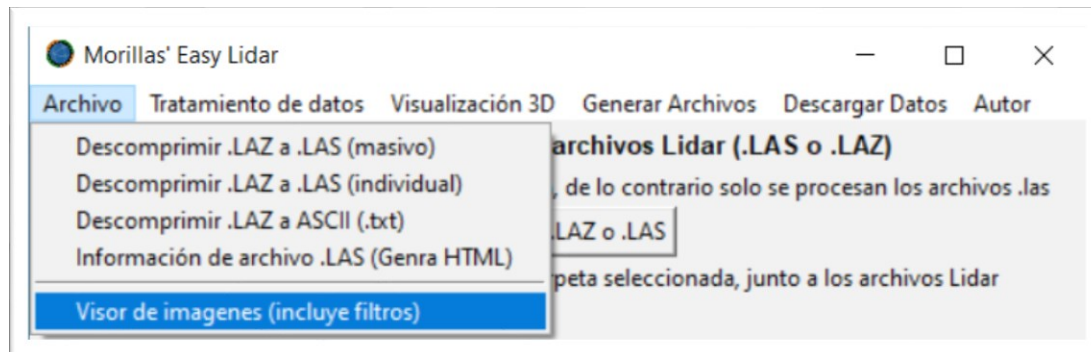


Figura 8.23. Opción para lanzar el visor/editor de imágenes desde el menú Archivo
Fuente: elaboración propia

8.3.1. Lenguajes, librerías y programas de terceros

El software de escritorio desarrollado para la presente tesis ha sido programado en Python 3.7.1. Para la generación de la interfaz se ha utilizado la librería Tkinter, para el procesado de los datos, la librería Whitebox y, por último, para la integración de utilidades desarrolladas en otros lenguajes, se ha usado la librería “Miscellaneous operating system interfaces (OS)”

Además de las librerías anteriormente mencionadas, se han integrado en el software funciones obtenidas a través de llamadas mediante la consola de comandos a software de terceros, éstos son LAsTools de rapidlasso GmbH e IrfanView de Irfan Skiljan.

8.4. Fuentes de datos

Durante la presente investigación se han utilizado diversas fuentes de datos, en función de su aplicación en cada una de las herramientas desarrolladas, accediendo a los datos bajo demanda, para la herramienta online y las hojas de nube de puntos (archivos .laz), para la herramienta de escritorio.

8.4.1. Fuentes de datos del software de realidad virtual desarrollado para su uso online

En primer lugar, para los modelos tridimensionales que se muestran en la herramienta online, se han utilizado, principalmente, los datos del MDE suministrado por Cesium. Estos modelos son cubiertos con las texturas procedentes de las distintas capas, principalmente WMS (Web Map Service). Los datos, tanto del terreno como de las texturas aplicadas al mismo, se obtienen mediante solicitudes HTTP REST a los distintos servidores.

Para tener una mayor aproximación al geoide terrestre se ha incluido de forma predeterminada la capa de terreno, basada en MDE “Cesium World Terrain”, elaborado por Cesium.ion, la cual carga un modelo tridimensional de la tierra basado en los datos del Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia.

Para mejorar la eficiencia de la carga del software en dispositivos con menor capacidad de procesamiento, se ha incluido la posibilidad de mantener la tridimensionalidad del globo terráqueo, eliminando los modelos digitales de elevaciones mediante la utilización de las opciones del menú. Esta representación tridimensional más ligera se basa en el elipsoide WGS84, la cual

se crea procedimentalmente. Esto carece de la apariencia realista del terreno, pero no solicita datos de un servidor, lo cual acelera la respuesta de los visores desarrollados en el proyecto. Para poder desactivar los datos de relieve y mostrar el elipsoide, debe desactivarse la opción de terreno, a través del menú de opciones.

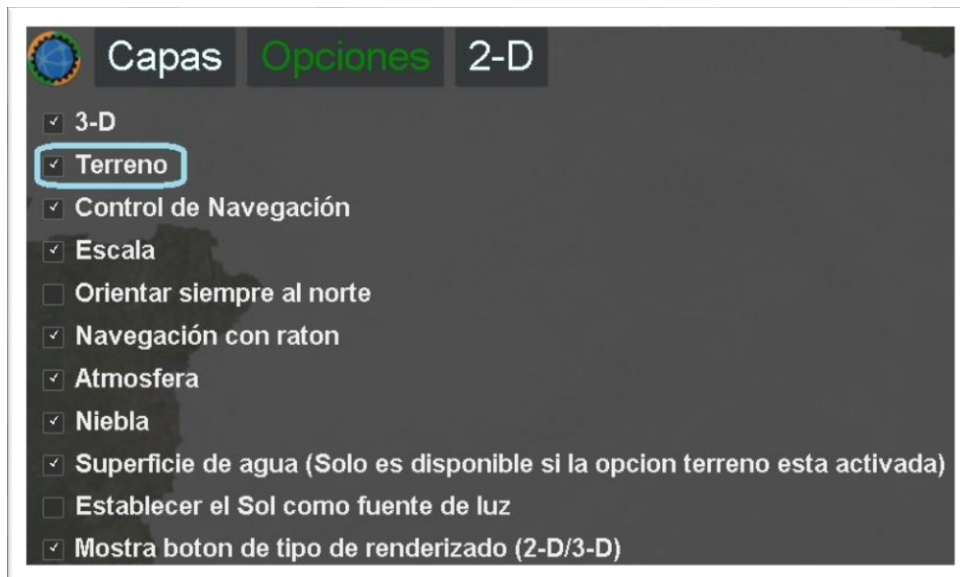


Figura 8.24. Imagen de la opción del menú del software online, para la activación o desactivación del relieve tridimensional del terreno
Fuente: elaboración propia, a partir de datos del IGN

Las teselas de terreno se obtienen del servidor en formato “gzipped”, consiguiendo así una mayor transmisión de datos y, por lo tanto, una respuesta más rápida. Una vez extraídos, los mosaicos son datos binarios little-endian.

La información primaria sobre altitud permite el análisis morfométrico con la precisión establecida por la resolución de la malla, la cual varía en función del territorio, debido a las diferentes fuentes de datos que se han utilizado para crear la malla.

La relación entre las diferentes zonas y la resolución de los datos disponibles viene recogida en la siguiente tabla:

Cobertura	Resolución aproximada
Inglaterra	1 m a 2 m
Australia (áreas pobladas y costas)	5 m
Nueva Zelanda	8 m
Costa oeste de estados unidos	50 cm
Estados Unidos	3 m a 30 m
Canadá	30 m
Europa	30 m
Méjico	30 m
Entre -60° a 60° de latitud	30–90 m
Tierra entera	1,000 m

Como se ha mencionado anteriormente, la malla está formada por distintos mosaicos con características distintas. Cada mosaico es una malla triangular especialmente codificada, donde los vértices se superponen con sus vecinos en los bordes de los mosaicos.

Los datasets de terreno son enormes, por lo general suman gigabytes o terabytes. Debido a esto, se ha escogido la API suministrada por Cesium, para llevar a cabo la consulta de datos para alimentar el SIG de Realidad Virtual.

Para poder llevar a cabo correctamente la solicitud de datos a la API de Cesium, debe realizarse un HTTP request mediante JavaScript, a través de esta consulta, a medida que se realiza zoom sobre el terreno; el sistema solicita un terreno de mayor resolución en función de las partes del globo que se muestran

y la distancia a la que se encuentran.

El servidor que provee el terreno incluye datos de costa necesarios para aguas animadas. También se han incluido datos para la iluminación del terreno, así como datos de la costa necesarios para los efectos del agua. Las olas se animan en base a la fecha y la hora, y muestran el reflejo del sol y la luna correspondientes al momento actual.

Para habilitar la iluminación del terreno, se solicita la extensión VertexNormals del servidor de terreno y se habilita la iluminación en el escenario tridimensional; este proceso se realiza de forma automática mediante la GUI (Interfaz Gráfica de Usuario).

Para todas las capas dispuestas en los visores online, el sistema de coordenadas empleado es el Sistema Geodésico Mundial (WGS84) y el sistema de referencia es EPSG: 4326.

Las capas que se han incluido en el visor son las siguientes:

- Nomenclátor Geográfico Básico de España
- Inventario Español de Lugares de Interés Geológico (IELIG)
- MDT - Mapa del Relieve (5m)
- MDT - Hillshade (5m) MDT-Pendientes (25m)
- MDT - Orientaciones (25m) PNOA Máxima actualidad
- Mapa Geológico de España 1:50.000 (MAGNA)
- Minutas Cartográficas entre 1870 y 1950
- Primera edición MTN25
- Zonas GEODE - Mapa Geológico Continuo de España a escala 1:50.000
- Cuaternario - Recintos-Mapa Geológico Continuo de España a escala 1:50.000
- Cuaternario - Líneas auxiliares - Mapa Geológico Continuo de España a escala 1:50.000

- Buzamientos - Recintos-Mapa Geológico Continuo de España a escala 1:50.000
- Recintos geología - Mapa Geológico Continuo de España a escala 1:50.000
- Mapa de Estructuras de plegamiento 1:50.000 (MAGNA 50)
- Mapa de Contactos y fallas 1:50.000 (MAGNA 50)
- Mapa de Litologías tramas 1:50.000 (MAGNA 50)
- Litologías color 1:50.000 (MAGNA 50)
- Contactos - Mapa Geológico Continuo de España a escala 1:50.000
- Mapa Geológico 1:200.000
- Litologías color - Mapa Geológico 1:200.000
- Litologías tramas - Mapa Geológico 1:200.000
- Contactos y fallas - Mapa Geológico 1:200.000
- Estructuras de plegamiento - Mapa Geológico 1:200.000
- Litoestratigrafía - Mapa Litoestratigráfico de España a escala 1:200.000
- Contactos y fracturas - Mapa Litoestratigráfico de España a escala 1:200.000
- Estructuras - Mapa Litoestratigráfico de España a escala 1:200.000
- Mapa Litologías 1:1.000.000
- Mapa de Edades de España 1:1.000.000
- Unidades tectónicas - Neotectónico de España a escala 1:1.000.000
- Estructuras - Neotectónico de España a escala 1:1.000.000
- Macroestructuras - Neotectónico de España a escala 1:1.000.000
- Isobatas - Neotectónico de España a escala 1:1.000.000
- Mesoestructuras - Neotectónico de España a escala 1:1.000.000
- Unidades tectónicas - Tectónico 1:1.000.000
- Unidades metamorfismo - Tectónico 1:1.000.000
- Karst de España a escala 1:1.000.000
- Estructuras de plegamiento - Tectónico 1:1.000.000
- Contactos y Fallas - Tectónicas 1:1.000.000
- Volcanes - Mapa Geológico 1:1.000.000 del año 2015

- Diques - Mapa Geológico 1:1.000.000 del año 2015
- Formas Canarias - Mapa Geológico 1:1.000.000 del año 2015
- Estructuras Plataforma - Mapa Geológico 1:1.000.000 del año 2015
- Estructuras Península Ibérica y Baleares - Mapa Geológico 1:1.000.000 del año 2015
- Unidades Geológicas Península Ibérica, Baleares, Canarias, Madeira y Azores Mapa Geológico 1:1.000.000 del año 2015
- Unidades Geológicas Plataforma Islas Canarias - Mapa Geológico 1:1.000.000 del año 2015
- Unidades Geológicas Plataforma Islas Península Ibérica y Baleares - Mapa Geológico 1:1.000.000 del año 2015
- Escarpes - Base de Datos de Movimientos del terreno (BDMOVES)
- Masas Movidas - Base de Datos de Movimientos del terreno (BDMOVES)
- Zonas con Movimientos del Terreno Generalizados
- Base de Datos de Movimientos del terreno (BDMOVES)
- Flujos de derrubios - Base de Datos de Movimientos del terreno (BDMOVES)
- Eventos observados - Base de Datos de Movimientos del terreno (BDMOVES)
- Base de Datos de Fallas del Cuaternario de Iberia (QAFI)
- Zonas sismogénicas de la Península Ibérica y territorios de influencia 1:50.000 (MAGNA 50)
- Medidas estructurales anotaciones 1:50.000 (MAGNA 50)
- Identificadores de las unidades cartográficas anotaciones 1:50.000 (MAGNA 50)
- Identificadores de las unidades cartográficas líneas 1:50.000 (MAGNA 50)
- E R Puntual 1:50.000 (MAGNA 50)
- E R Puntual Tectónico-Tectónico 1:1.000.000
- Medidas estructurales 1:50.000 (MAGNA 50)
- E O Diverso líneas 1:50.000 (MAGNA 50)

- E O Diverso-Tectónico 1:1.000.000
- E O Diverso polígonos 1:50.000 (MAGNA 50)
- Ejes - Recintos-Mapa Geológico Continuo de España a escala 1:50.000
- Indicios minerales - Mapa Geológico Continuo de España a escala 1:50.000
- Líneas auxiliares - Mapa Geológico Continuo de España a escala 1:50.000
- Rótulos unidades - Mapa Geológico Continuo de España a escala 1:50.000
- PENOA-2017
- PENOA-2016
- PENOA-2015
- PENOA-2013
- PENOA-2012
- PENOA-2011
- PENOA-2010
- PENOA-2009
- PENOA-2008
- PENOA-2007
- PENOA-2006
- PENOA-2005
- PENOA-2004
- SIGPAC (1997-2003)
- OLISTAT (1997-1998)
- Vuelo Nacional (1981-1986)
- Vuelo Interministerial (1973-1986) Vuelo americano (Serie B, 1956-1957)
- Años de los vuelos PNOA
- Base de Datos de Metalogenia (Yacimiento e Indicios Minerales)
- Geológico de La Palma a escala 1:100.000
- Geológico de Menorca a escala 1:100.000
- Geológico de Ibiza y Formentera a escala 1:100.000

- USGS Imagen Mundial
- Topografía Mundial
- Hillshade del Relieve Mundial
- Espectro Visible (15 min)
- Espectro Visible (1 hora)
- Vapor de agua (1 hora)
- Infrarrojo (1 hora)
- Infrarrojo (CONUS, 15 min)

A continuación, se especifican algunos detalles de las fuentes de datos utilizadas para crear las capas anteriormente expuestas.

8.4.2. Capas para el estudio del relieve

MDT	Datos capturados con sensores LiDAR aerotransportados del proyecto PNOA-LiDAR con paso de malla de 5m del Sistema Cartográfico Nacional (http://www.scne.es/productos.html) y los mapas de pendientes, orientaciones, sombreado y relieve, procedentes de otras fuentes. Servicio de visualización WMS 1.3.0 conforme al perfil Inspire de ISO 19128:2005. Procedencia de los datos: Instituto Geográfico Nacional.
MDT-Mapa del Relieve (5m)	Modelo Digital del Terreno que representa el relieve del terreno. Paso de malla: 5 metros. Ámbito geográfico: España peninsular, Islas Baleares, Ceuta y Melilla. Origen de datos: Instituto Geográfico Nacional.
MDT-Hillshade	Sombreado del relieve generado a partir del Modelo Digital del Terreno de paso de malla 5 metros. Ámbito geográfico: España peninsular, Islas Baleares, Islas Canarias, Ceuta y Melilla. Origen de datos: Sistema Cartográfico Nacional (http://www.scne.es)

MDT-Orientaciones	Modelo Digital del Terreno que representa las orientaciones del terreno. Paso de malla: 25 metros. Ámbito geográfico: España peninsular, Islas Baleares, Ceuta y Melilla. Origen de datos: Instituto Geográfico Nacional. Rango de escalas de visualización: escalas menores a 1:60.000.
MDT-Pendientes	Modelo Digital del Terreno que representa las pendientes del terreno. Paso de malla: 25 metros. Ámbito geográfico: España peninsular, Islas Baleares, Ceuta y Melilla. Origen de datos: Instituto Geográfico Nacional. Rango de escalas de visualización: escalas menores a 1:60.000.
Inventario Español de Lugares de Interés Geológico (IELIG):	Base de datos del Inventario Español de Lugares de Interés Geológico (IELIG) que, de acuerdo con la Ley 42/2007, debe elaborar y actualizar el Ministerio de Agricultura, Alimentación y Medio Ambiente, con la colaboración de las Comunidades Autónomas y de las instituciones de carácter científico. El Real Decreto 1274/2011, encomienda al IGME la finalización de este inventario, sin perjuicio de las actuaciones que las Comunidades Autónomas, en uso de sus competencias, lleven a cabo para completarlo en sus respectivos territorios.
PENOA Máxima Actualidad	Cobertura ráster opaca de imágenes de satélite y ortofotos PNOA de Máxima Actualidad (MA). Rangos de visualización: Imagen Spot5 de 10 m de resolución hasta la escala aproximada 1:70 000; a partir de aquí ortofotografías PNOA MA de 0.25m o 0.50m de resolución, según la zona.

8.4.3. Capas para el estudio de la geología

IGME - Mapa Geológico de España 1/50.000 (MAGNA)	El Mapa Geológico Nacional (MAGNA), realizado entre 1972 y 2003 por el Instituto Geológico y Minero de España, se distribuye en hojas 1:50.000 (1:25.000 en los casos particulares de Canarias, Ibiza-Formentera, Menorca, Ceuta y Melilla). En el Mapa Geológico se representa la naturaleza de los materiales
---	---

	(rocas y sedimentos) que aparecen en la superficie terrestre, su distribución espacial y las relaciones geométricas entre las diferentes unidades cartográficas. Como capas derivadas de este mapa, se han incluido las siguientes: Litologías color, Litologías tramas, Contactos y fallas, Estructuras de plegamiento, E O Diverso polígonos, E O Diverso líneas, Medidas estructurales, E R Puntual, Identificadores de las unidades cartográficas líneas, Identificadores de las unidades cartográficas anotaciones, Medidas estructurales anotaciones y Hojas 1:50.000.
IGME - Mapa Geológico Continuo de España a escala 1/50.000	El Mapa Geológico de España Continuo a escala 1:50.000 para la Península y 1:25.000 en los territorios insulares. Incluye las unidades cronolitoestratigráficas separadas por distintos tipos de contactos. Las capas derivadas de este mapa, que han sido incluidas en el visor son las siguientes: Zonas GEODE, Recintos geología, Contactos, Cuaternario - Recintos, Cuaternario - Líneas auxiliares, Ejes, Buzamientos, Indicios minerales, Líneas auxiliares y Rótulos unidades
Mapa Geológico de España a escala 1:200.000	Mapa con las unidades cronolitoestratigráficas de la zona cubierta por la hoja, separadas por distintos tipos de contactos: normal o concordante, discordante, intrusivo y de otra índole. La estructura tectónica se representa mediante las trazas cartográficas de los pliegues, la orientación y buzamiento de los elementos planares y lineares contenidos en las rocas, así como por la relación estructural (mediante fallas y cabalgamientos) entre los conjuntos rocosos diferenciados cartográficamente. También se representan características singulares de las rocas ígneas, así como el grado de migmatización de las rocas metamórficas. Se han incluido en el visor online, las siguientes capas derivadas de este mapa: Litologías color, Contactos y fallas, Litologías tramas y Estructuras de plegamiento.
Mapa Litoestratigráfico de España a escala 1:200.000	El Mapa Litoestratigráfico de España continuo y en formato digital a escala 1:200.000 representa las unidades cartográficas caracterizadas según criterios litoestratigráficos e hidrogeológicos fundamentalmente. Los criterios seguidos para considerar su representación cartográfica son el mantenimiento

	<p>de aquellas unidades que presentan un importante desarrollo litoestratigráfico, tanto por su representación como por su singularidad, incorporando además las que manifiestan un alto interés hidrogeológico, bien por su propia naturaleza litológica (alta permeabilidad) o porque se consideran que forman parte esencial de la definición del acuífero. Los contactos entre las unidades cartográficas representados fueron los contactos normales, discordantes e intrusivos. Se representan fallas, cabalgamientos y estructuras de plegamiento. Así, se ha obtenido un mapa continuo y en formato digital, en el que persisten zonas con una densidad de información cartográfica mayor que otras debido, bien a que proceden de mapas de síntesis realizados con criterios menos generalistas que el de estas, o bien a que para su confección se ha recurrido a la utilización de cartografías a menor escala (p. e. 1:50.000), obteniéndose así una cartografía mucho más rica y detallada. Este mapa es la base de los mapas de permeabilidades e hidrogeológico de España a escala 1:200.000 continuos y en formato digital. Esta cartografía está adaptada a la red hidrográfica y al modelo de sombras realizado por el CEDEX, a escala 1:200.000, a partir de la base topográfica, a escala 1:50.000, del Servicio Cartográfico del Ejército. A partir de este mapa se han creado las capas de litoestratigrafía, contactos y fracturas, y estructuras.</p>
<p>Mapa Tectónico de la Península Ibérica y Baleares escala 1:1.000.000</p>	<p>Mapa de distribución geográfica de las grandes unidades estructurales de la Península Ibérica y Baleares, clasificadas en dos grandes grupos: áreas de plataforma y orógenos, de los que han derivado tres unidades: Cordillera Hercínica, Cordilleras Alpidicas y Áreas de Plataforma, Mesozoicas y Terciarias. Las unidades cartográficas representadas se han delimitado a partir de sus relaciones temporales con la edad de la deformación, en el caso de las Cordilleras. Las relaciones estructurales entre las unidades quedan reflejadas por la traza de las grandes estructuras de plegamiento y las fallas. También se incluyen los distintos grados de metamorfismo al que se han visto sometidos los diferentes grupos rocosos. Las capas derivadas de este mapa que han sido incluidas en el visor son:</p> <p>Unidades tectónicas.</p>

	<p>Unidades de metamorfismo.</p> <p>E O Diverso Tectónico.</p> <p>Estructuras de plegamiento Tectónico.</p> <p>Contactos y Fallas Tectónicas.</p> <p>E R Puntual Tectónico.</p>
<p>Mapa Neotectónico de España a escala 1:1.000.000</p>	<p>El Mapa Neotectónico expresa aquellos rasgos estructurales, morfológicos, estratigráficos y de todo tipo, que ponen en evidencia movimientos relativamente recientes (de la época que se ha considerado neotectónica), así como el tipo de esfuerzo al que está sometida la península. Sus objetivos son la identificación de las deformaciones desde el límite Mioceno Medio - Mioceno Superior hasta la actualidad. Los materiales se han clasificado en seis categorías, dos para materiales volcánicos diferenciados por edad; y cuatro para el resto de los materiales, con cuatro subdivisiones en función de la edad. También comprende estructuras tectónicas, direcciones de esfuerzo y deformación, estructuras diapíricas y otros fenómenos relacionados, todos con expresión de su edad. El mapa contiene información de la plataforma continental española. Las capas derivadas de este mapa, que han sido incluidas en el visor online, son: Unidades tectónicas, Estructuras, Macroestructuras Neotectónico, Isobatas Neotectónico y Mesoestructuras.</p>
<p>Mapa Geológico de España y Portugal a escala 1:1.000.000 del año 2015</p>	<p>Es una versión cartográfica actualizada de la Península Ibérica, así como de Ceuta y Melilla y de todo el territorio insular de España y Portugal: Islas Baleares, Canarias, Azores y Madeira, editada de forma periódica por el IGME en colaboración con el Servicio Geológico de Portugal (LNEG). En esta nueva versión se incluye, por primera vez, la cartografía geológica de la plataforma continental y de las islas atlánticas de Portugal. En este mapa se recoge todo el nuevo conocimiento geológico que ha supuesto la finalización de la serie MAGNA de cartografía geológica escala 1:50.000 del IGME, para el territorio español y de las series sistemáticas de cartografía del territorio de Portugal, desarrolladas por el LNEG. La parte continental del mapa se ha compuesto a partir de síntesis previas a escala 1:400.000 desarrolladas para cada gran unidad geológica de</p>

	<p>España, concretamente, Macizo Varisco, Pirineos, C. Bética, C. Cantábrica e Ibérica y Cuencas Cenozoicas del Duero, Tajo y Ebro, y en la cartografía geológica a escala 1:200.000 producida en Portugal. Las capas derivadas de este mapa que se han incluido son las siguientes: Unidades Geológicas Plataforma Islas Península Ibérica y Baleares, Unidades Geológicas Plataforma Islas Canarias, Unidades Geológicas Península Ibérica, Baleares, Canarias, Madeira y Azores, Estructuras Península Ibérica y Baleares, Estructuras Plataforma, Formas Canarias, Diques y Volcanes.</p>
<p>Base de Datos de Fallas del Cuaternario de Iberia (QAFI)</p>	<p>Muestra las trazas cartográficas de las fallas que forman parte la Base de Datos de Fallas Activas en el Cuaternario de la Península Ibérica.</p>
<p>Base de datos de zonas sismogénicas de la Península Ibérica y territorios de influencia desarrollada para el cálculo de la actualización del mapa de peligrosidad sísmica de España (IGN-UPM, 2013)</p>	<p>ZESIS es una base de datos de zonas sismogénicas de la Península Ibérica y territorios de influencia desarrollada para el cálculo de la actualización del mapa de peligrosidad sísmica de España (IGN-UPM, 2013). ZESIS es el resultado de la evolución de tres modelos sucesivos en los que han colaborado numerosos investigadores tanto de centros de investigación nacionales como internacionales, y de la sinergia de los proyectos FASEGEO (CGL2009-09726), SHARE (FP7-226967), IBERFAULT (CGL2009-07388), OPPEL (IGN-UPM) y SISMOGEN (IGME).</p>
<p>Base de Datos de Movimientos del terreno (BDMOVES)</p>	<p>Base de datos de los movimientos del terreno (deslizamientos, subsidencias y colapsos) que han sido inventariados por el IGME. Se han incluido las capas derivadas: Masas Movidas, Zonas con Movimientos del Terreno Generalizados, Flujos de derrubios, Escarpes, Eventos Observados</p>

8.4.4. Ortofotos Históricas

Imágenes ortorectificadas de distintos vuelos fotográficos de cobertura nacional promovidos por distintas Administraciones Públicas. Servicio WMS 1.3.0 conforme a ISO 19128:2005.

Cobertura de ortoimágenes correspondientes al proyecto PNOA (por años)	La cobertura PNOA está constituida por mosaicos de distinta resolución (50 y 25 cm). Se incluye una capa vectorial para consultar los vuelos PNOA disponibles en cada punto.
Años de los vuelos PNOA	<p>Esta capa permite consultar los años de los vuelos PNOA disponibles en cada punto. Se ofrecen dos estilos de visualización.</p> <p>PNOA-2017: Ortoimágenes del PNOA correspondientes al año 2017.</p> <p>PNOA-2016: Ortoimágenes del PNOA correspondientes al año 2016.</p> <p>PNOA-2015: Ortoimágenes del PNOA correspondientes al año 2015.</p> <p>PNOA-2014: Ortoimágenes del PNOA correspondientes al año 2014.</p> <p>PNOA-2013: Ortoimágenes del PNOA correspondientes al año 2013.</p> <p>PNOA-2012: Ortoimágenes del PNOA correspondientes al año 2012.</p> <p>PNOA-2011: Ortoimágenes del PNOA correspondientes al año 2011.</p> <p>PNOA-2010: Ortoimágenes del PNOA correspondientes al año 2010.</p> <p>PNOA-2009: Ortoimágenes del PNOA correspondientes al año 2009.</p> <p>PNOA-2008: Ortoimágenes del PNOA correspondientes al año 2008.</p> <p>PNOA-2007: Ortoimágenes del PNOA correspondientes al año 2007.</p> <p>PNOA-2006: Ortoimágenes del PNOA correspondientes al año 2006.</p> <p>PNOA-2005: Ortoimágenes del PNOA correspondientes al año 2005.</p> <p>PNOA-2004: Ortoimágenes del PNOA correspondientes al año 2004.</p>
SIGPAC (1997-2003)	Ortoimágenes del vuelo SIGPAC realizadas entre 1997 y 2003 sobre la totalidad del territorio español. El vuelo fue realizado por el Ministerio de Agricultura, a través del Fondo Español de Garantía Agraria (FEGA), junto con aportaciones de algunas CCAA, con el objetivo de generar las ortofotos que sirvieran de referencia para el Sistema de Identificación

	de Parcelas Agrícolas (SIGPAC). Esta capa permite consultar las fechas de vuelo y la resolución de la ortofoto en cada zona.
OLISTAT (1997-1998)	Ortoimágenes del vuelo OLISTAT Oleícola realizado por el Ministerio de Agricultura entre 1997 y 1998 sobre parte del territorio español para contabilizar el número de olivos del territorio español. Permite consultar las fechas de vuelo de cada zona. La resolución de la ortofoto es de un metro en todo el vuelo.
Vuelo Nacional (1981-1986)	Ortoimágenes del vuelo nacional realizado entre 1981 y 1986 sobre la totalidad del territorio español a escala 1:30.000 y realizado por empresas privadas contratadas por el IGN. Esta capa permite consultar la resolución de la ortoimagen y la fecha de vuelo.
Vuelo Interministerial (1973-1986)	Ortoimágenes del vuelo realizado entre noviembre de 1976 y julio de 1986 por el Ministerio de Agricultura, el M.O.P.U , Hacienda y el Ministerio del Aire. Esta capa permite consultar la resolución de la ortoimagen y la fecha de vuelo de cada fotograma.
Vuelo americano (Serie B, 1956-1957)	Ortoimágenes del vuelo realizado por el Army Map Service de EEUU entre enero de 1956 y noviembre de 1957 sobre parte del territorio español, conocido como Vuelo Americano, serie B. Esta capa permite consultar la resolución de la ortoimagen.
MTN25 Primera edición	Primera edición del Mapa Topográfico Nacional (MTN) y las minutas obtenidas por métodos fotomecánicos. A través de la operación GetFeatureInfo se puede acceder a todas las ediciones disponibles de cada hoja MTN50 o MTN25 en el centro de descargas del CNIG o bien a información sobre las Minutas. Servicio WMS 1.3.0 conforme a ISO 19128:2005.
Mapas ráster del IGN	Cartografía ráster del IGN a distintas escalas. Se publica el Mapa de España a escala 1:2.000.000 hasta una resolución de 420 m/píxel. El Mapa de España a escala 1:1.250.000 hasta una resolución de 104,44 m/píxel. El Mapa de España a escala 1:500.000 hasta una resolución de 40,04 m/píxel. El Mapa Provincial a escala 1:200.000 hasta una resolución de 20,16 m/píxel. El Mapa Topográfico Nacional a escala 1:50.000 hasta una resolución de 5,04 m/píxel y el Mapa Topográfico Nacional a escala 1:25.000 a partir de una resolución de 5,04 m/píxel. Se considera un tamaño estándar de píxel de 0,28mm. Servicio WMS 1.3.0 conforme a ISO 19128:2005.

Nombres geográficos	<p>Representación de nombres geográficos como los nombres de zonas, regiones, localidades, ciudades, periferias, poblaciones o asentamientos, o cualquier rasgo geográfico o topográfico de interés público o histórico procedentes del Nomenclátor Geográfico Básico de España (NGBE). Sus denominaciones son conformes con la Directiva Inspire 2007/2/EC, así como su estilo. Además, tienen asociado un estilo de visualización definido por el IGN. Servicio de visualización WMS 1.3.0 conforme al perfil Inspire de ISO 19128:2005.</p> <p>Las entidades recogidas son: Alineación montañosa, Montaña, Paso de montaña, Llanura, Depresión, Vertiente, Entrante costero y estrecho marítimo, saliente y otro relieve costero. Reglamento (UE) N.º 1089/2010.</p>
Minutas Cartográficas	<p>Capa creada a partir de archivos ráster correspondientes a la digitalización de los mapas manuscritos en papel conservados en el Archivo Técnico del IGN. Se trata de los trabajos previos a la realización del Mapa Topográfico Nacional, en algunos casos con varias décadas de diferencia a la publicación de la primera edición del MTN de la zona. Este tipo de documentos se realizaron principalmente entre 1870 y 1950 y se clasifican en minutas planimétricas, minutas altimétricas y minutas conjuntas de altimetría y planimetría. Dibujados a escala 1:25.000, con una precisión de obtención de la información correspondiente a escala 1:50.000. A través de la operación GetFeatureInfo se puede acceder a minutas planimétricas y minutas altimétricas disponibles en el centro de descargas del CNIG.</p>
Microsoft Satellite	<p>Capa ráster basada en ortoimágenes satelitales y aéreas ofrecidas por Microsoft a través de Bing Maps. En muchas áreas, la resolución máxima es de aproximadamente 4,5 píxeles por metro. En otros lugares, especialmente en las zonas más remotas del mundo, la resolución disponible es unas pocas órdenes de magnitud menor.</p>
Microsoft Satellite con etiquetas	<p>Posee las mismas características que la capa anterior, pero se han añadido etiquetas que identifican los lugares.</p>
Microsoft Carreteras	<p>Mapa de carreteras ofrecido por Microsoft, con características similares a las anteriores.</p>
Mapbox Satellite	<p>Imágenes satelitales y aéreas globales de proveedores comerciales, principalmente DigitalGlobe; Nearmap; Geodatastyrelsen, FOT</p>

	Orthophoto, 2014; LGV Hamburg; National Land Survey of Finland, NLS Orthophotos, 2014; LINZ Data Service (CC BY 3.0 NZ); Geoportal Berlin, DOP orthophotos, 2014; Town of Banff Open Data Imagery; City of Kamloops Open Data Imagery (Open Government License - Kamloops); NASA; USDA; and USGS.
Mapbox Carreteras	Capa vectorial de carreteras ofrecida por Mapbox y basadas principalmente en Open Street Maps.
Esri National Geographic	<p>Capa ráster geográfica multiescala, suministrada por ESRI, a partir de los datos de diversos proveedores, principalmente Garmin, HERE, UNEP-WCMC, NASA, ESA, USGS. La escala de esta capa es aproximadamente 1:144.000 o menor para la mayoría de los territorios, siendo la de Norteamérica la más detallada, con una escala inferior en la mayoría de los emplazamientos a 1:9.000.</p> <p>La escala y resolución de las últimas 6 capas anteriormente mencionadas cambia en función del lugar, y sus fuentes son mejoradas con gran frecuencia, por lo que no se puede establecer una resolución específica para dichas fuentes.</p>

8.4.5. Capas para teledetección atmosférica

Las capas orientadas a la teledetección que se han incluido en el visor proceden del proyecto NOAA-NASA GOES, a partir de los datos recogidos por el satélite GOES, con actualizaciones periódicas cada 15 minutos o una hora según la capa y tienen una resolución aproximada de 4 km/pixel. Las imágenes son suministradas por UW-Madison SSEC.

Las capas incluidas son:

- Espectro Visible (15 min)
- Espectro Visible (1 hora)
- Vapor de agua (1 hora)
- Infrarrojo (1 hora)
- Infrarrojo (CONUS, 15 min)

8.5. Fuentes de datos del software MEL

Los ficheros digitales de la nube de puntos LiDAR son distribuidos en ficheros de 2x2 km de extensión, con formato de archivo .laz (.las comprimido). El sistema geodésico de referencia de estos archivos es ETRS89 (en las Islas Canarias es REGCAN95), siendo ambos sistemas compatibles con WGS84 y su proyección es UTM en el huso correspondiente a cada fichero. En algunos casos los datos son servidos en dos husos si la extensión de los datos abarca territorios con dos husos distintos.

Alturas ortométricas. Las nubes de puntos han sido capturadas mediante vuelo con sensor LiDAR. La densidad de puntos es de 0,5 puntos/m² en la primera cobertura y de 1 punto/m² en la segunda cobertura. La precisión altimétrica obtenida es de, al menos, 20 cm para el error cuadrático medio (RMSE) en la primera cobertura y de 15 cm en la segunda. El paso de malla en ambos casos es de 5 x 5 metros. Los ficheros han sido clasificados de manera automática, en base a la intensidad de la señal láser recibida, y coloreadas mediante RGB, partiendo de los valores de las ortofotos del Plan Nacional de Ortofotografía Aérea (PNOA) con tamaño de píxel variable, en función del año en el que se recogieron los datos y la zona. Algunos de los últimos lotes elaborados están en Infrarrojo y sin clasificar. Las coordenadas Altimétricas de estos puntos han sido transformadas en coordenadas Ortométricas, para lo cual se ha utilizado el modelo de geoide EGM2008-REDNAP (Adaptación del geoide mundial EGM08 a España).

8.5.1. Otras fuentes de datos

Para el estudio del relieve realizado, se han utilizado, en conjunción con los datos LiDAR del IGN, principalmente el Mapa Geológico de España Continuo a escala 1:50.000 para la península y 1:25.000 en los territorios insulares, el cual incluye las capas mostradas en la imagen inferior (Figura 8.25).

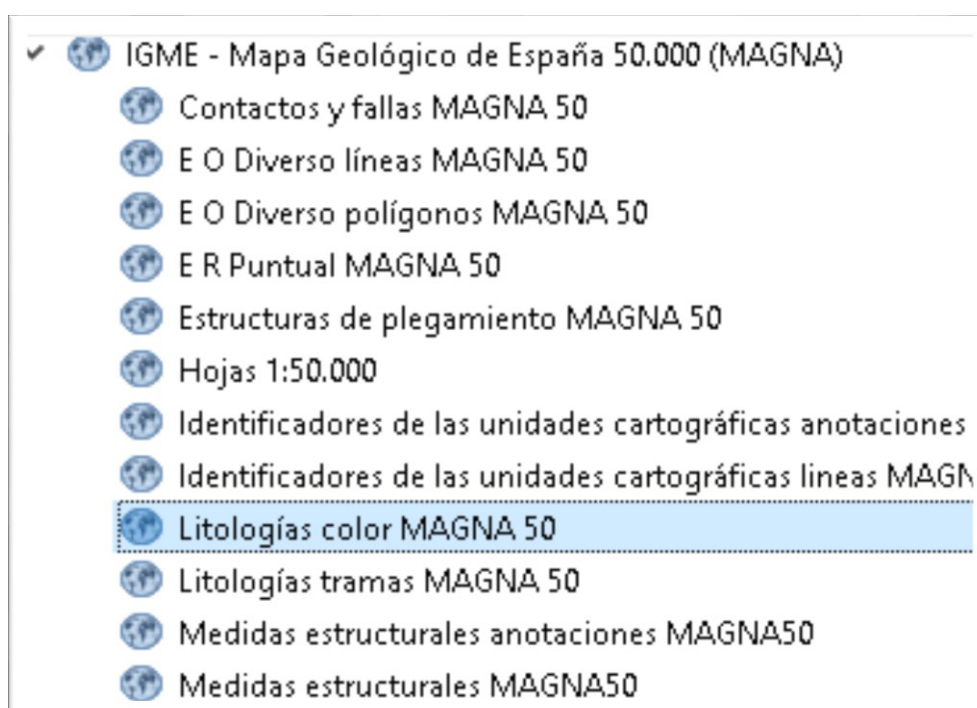


Figura 8.25. Imagen de capas del Mapa Geológico de España cargadas en QGIS
Fuente: elaboración propia

Los sistemas de referencia compatibles con el servicio que ofrece el IGME son CRS:84, EPSG:4326, EPSG:3857, EPSG:102100, EPSG:4258, EPSG:4230, EPSG:23028, EPSG:23029, EPSG:23030, EPSG:23031, EPSG:25828, EPSG:25829, EPSG:25831 y EPSG:25830.

9. Metodología de trabajo con datos 3D

Para poder realizar una limpieza y tratamiento de los datos tridimensionales, así como una correcta visualización y manipulación de éstos, en esta investigación se han creado dos herramientas: una online y otra de escritorio. Estas herramientas permiten el tratamiento de los datos y el estudio del relieve de forma más rápida y precisa que las herramientas existentes hasta ahora.

En cuanto a las herramientas elaboradas, en primer lugar, la herramienta online desarrollada es un SIG (<http://cartotecnia.com/teledeteccion/>), a través del cual puede estudiarse el relieve a nivel mundial y que permite, además, la utilización de sistemas de realidad virtual, para una mejor percepción del mismo. Este software, a su vez, permite la manipulación y superposición de las distintas capas que se han incluido, así como la activación y desactivación de determinados elementos para adecuarlo a las capacidades de los dispositivos desde los que a él se acceda, teniendo como finalidad, no sólo la utilización de éste para la realización del presente estudio del relieve, sino para poder ofrecer, de forma pública y gratuita, una herramienta que permita el análisis del relieve. Las instrucciones de uso vienen recogidas en este capítulo, mientras que el proceso de elaboración de este software está detallado en el capítulo 8.



Figura 9.1. Imagen de visor web con capa de edades a escala 1:1.000.000
Fuente: elaboración propia, a partir de los datos del IGN y del IGME

En segundo lugar, la herramienta de escritorio ha sido creada para el estudio más detallado de geomorfologías de menor tamaño y para la identificación de estructuras antrópicas, aprovechando al máximo la resolución de los datos de las nubes de puntos ofrecidas por el IGN. Esta herramienta, a la que se ha dado el nombre de MEL, ha sido diseñada para realizar el procesamiento masivo de los datos LiDAR, alimentándose con los datos directamente descargados del IGN, con extensión .laz. Este software permite al usuario realizar las operaciones que se realizan más habitualmente con nubes de puntos o modelos digitales del terreno, como la manipulación tridimensional de los modelos, el filtrado de los datos, la generación de hillshades, etc.

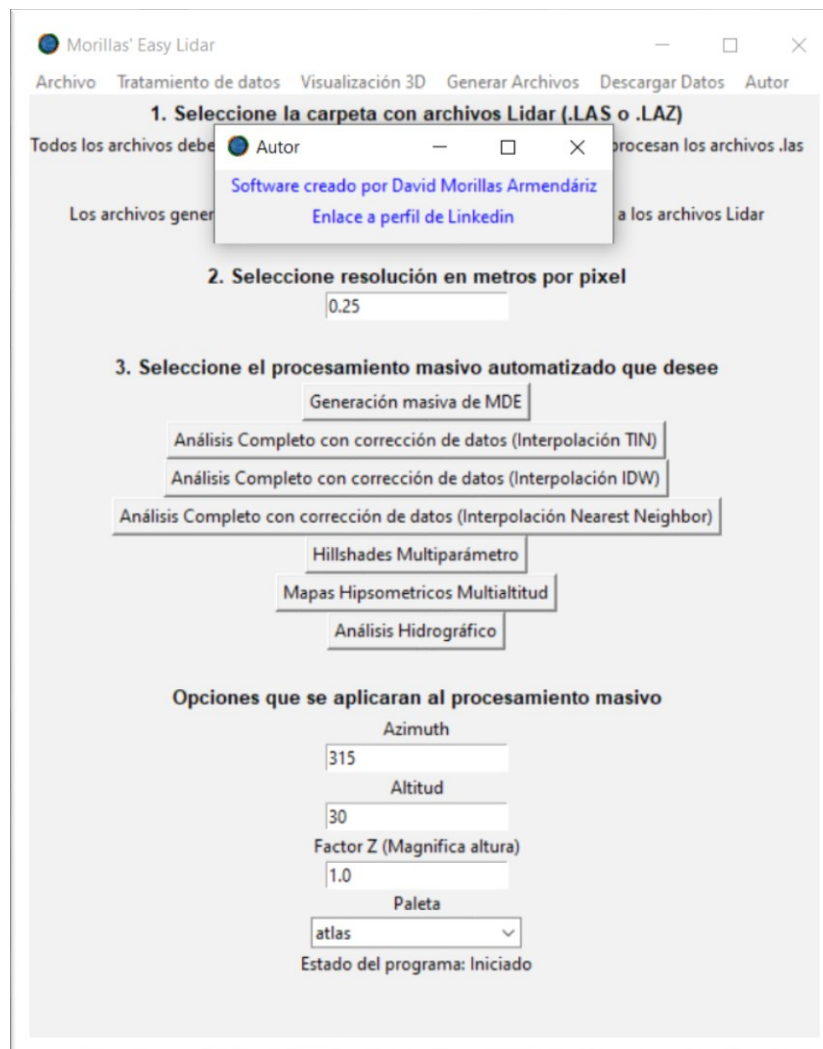


Figura 9.2. Créditos del software MEL
Fuente: elaboración propia

9.1. Descripción de la GUI (interfaz gráfica de usuario) de MEL y metodología de trabajo general

El software MEL ha sido creado para facilitar el estudio del relieve a través de nubes de puntos LiDAR y modelos digitales del terreno.

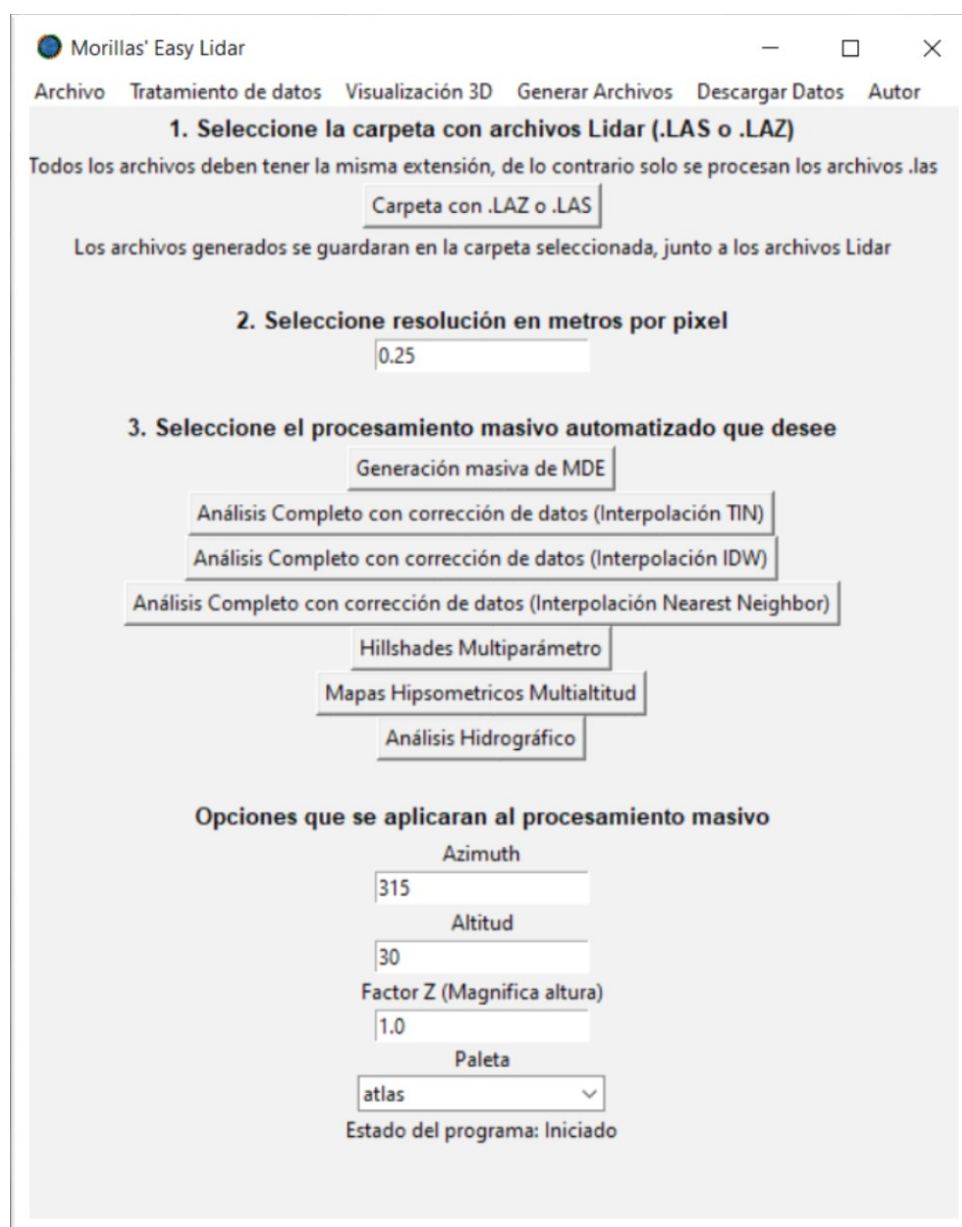


Figura 9.3. Interfaz principal del software MEL
Fuente: elaboración propia

Las diferentes opciones contenidas en la interfaz principal permiten el tratamiento masivo y automatizado de los datos, pudiendo éste realizarse con una intervención mínima del usuario. Para un uso básico de este software, basta con que el usuario seleccione la carpeta que contiene los archivos de nube de puntos.

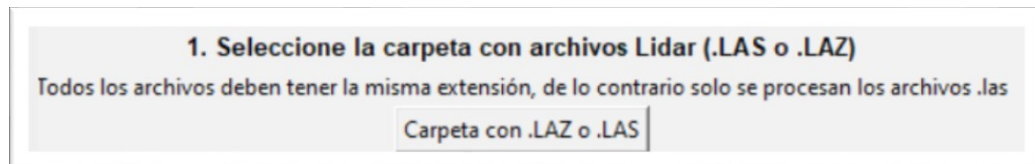


Figura 9.4. Detalle de la opción de selección de archivos para el tratamiento masivo de datos en la interfaz principal de MEL

Fuente: elaboración propia

Independientemente de la extensión del archivo, ya sea .LAZ o .LAS, el software MEL detectará de qué archivo se trata y procederá a la descompresión de éste si es necesario.

Una vez seleccionado el origen de los datos, basta con elegir la labor que se desea realizar haciendo clic en los procedimientos que se desean aplicar; todos ellos incluyen un pipeline que automatiza todos los procesos intermedios que habitualmente hay que realizar por separado y de forma manual, incluyendo el filtrado, limpieza y corrección de datos, así como el establecimiento de parámetros distintos para comprobar qué configuraciones son más adecuadas (generación de Hillshades multiparámetro y generación de mapas hipsométricos multialtitud).

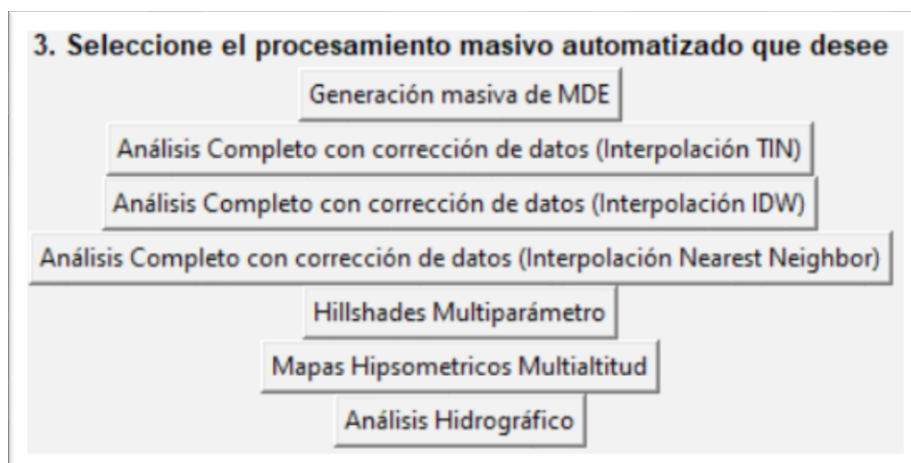


Figura 9.5. Detalle de las opciones de tratamiento masivo y automatizado de datos en la interfaz principal de MEL
Fuente: elaboración propia

De esta forma, con sólo unos pocos clics cualquier usuario puede procesar todos los datos LiDAR que desee y producir resultados con una resolución de 0.25 m por pixel.

Esta forma simple de procesar los datos utiliza las configuraciones paramétricas más habituales y las que han obtenido mejores resultados en las pruebas; sin embargo, el software MEL ofrece, además, la posibilidad de configurar los parámetros utilizados en la generación de archivos a varios niveles, dependiendo de los conocimientos del usuario. Para poder personalizar algunos de los parámetros fundamentales de los procesos automatizados mostrados en la Figura 9.5, se pueden modificar desde esta misma pantalla principal, la resolución de salida, el azimuth, la altitud, el factor z o la paleta de colores aplicada a los mapas hipsométricos.

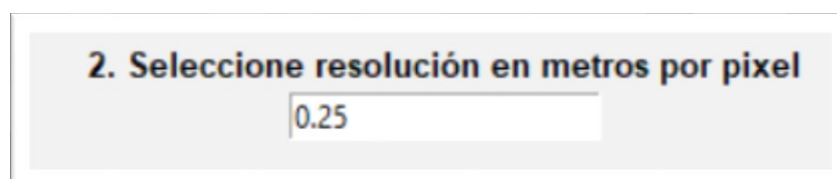


Figura 9.6. Cuadro de texto que permite modificar la resolución de los archivos generados automáticamente
Fuente: elaboración propia

Opciones que se aplicaran al procesamiento masivo

Azimuth
315

Altitud
30

Factor Z (Magnifica altura)
1.0

Paleta
atlas

Estado del programa: Iniciado

Figura 9.7. Opciones de configuración de los parámetros utilizados para la generación de hillshades y mapas hipsométricos de forma automatizada.

Fuente: elaboración propia

La resolución que se establezca afectará a todos los archivos generados de forma automatizada, mientras que los parámetros de azimuth, altitud, y factor z determinan las características de los hillshades generados de forma automatizada (salvo en el caso de la opción de “Hillshades Multiparámetro”, en la cual los hillshades generados se basan en distintas combinaciones de altitud y azimuth, con 30 grados de diferencia entre unas y otras), por los parámetros de paleta y altitud (este parámetro afecta tanto a los hillshades como a los mapas hipsométricos) afectarán al resultado de los mapas hipsométricos generados automáticamente (salvo en el caso de la opción de “Mapas Hipsométricos Multialtitud”, en el que se generan mapas hipsométricos con altitudes que van de 0 a 360 grados en intervalos de 15 grados).

Además del uso básico, el software MEL permite una serie de opciones más avanzadas, que cubren las necesidades más habituales que pueden surgir en el tratamiento de datos LiDAR y análisis de relieve. Para ello se han dispuesto en la barra superior una serie de menús, los cuales se detallan a continuación.

9.1.1. Herramientas avanzadas incluidas en MEL

Para poder realizar un análisis del relieve, el primer paso consiste en la adquisición de datos. Para facilitar esta tarea al usuario, se han incluido accesos directos a distintas fuentes de datos tanto nacionales como internacionales, a las que es posible acceder mediante el menú “Descargar Datos”.



Figura 9.8. Menú de descarga de datos
Fuente: elaboración propia

Una vez se dispone del archivo o los archivos LiDAR sobre los que se va a trabajar, lo conveniente es hacer una revisión preliminar del estado de los datos antes de proceder a su tratamiento; para ello se han incluido en el software varias herramientas de visualización que permiten comprobar no sólo la distribución tridimensional de los puntos contenidos en los archivos descargados, sino también verificar si han sido correctamente etiquetados y si su distribución es o no adecuada. Esta fase de visualización previa al tratamiento de los datos, también conocida como perfilado de datos es de gran importancia a la hora de determinar exactamente los procedimientos a los que se van a someter los datos; sin embargo, las opciones automatizadas vistas anteriormente incluyen procesos de limpieza, filtrado y corrección que son válidos para mejorar los datos y facilitar el análisis del relieve en la mayor parte de los casos.

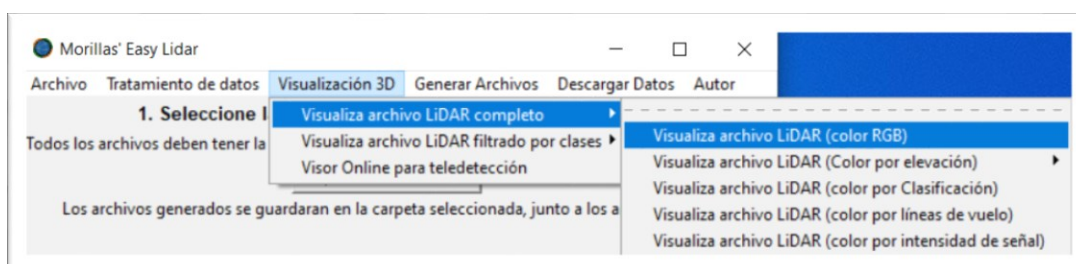


Figura 9.9. Submenú de visualización tridimensional de nube de puntos completa con distintas opciones de asignación de colores.

Fuente: elaboración propia

Dentro de las opciones de visualización tridimensional de nubes de puntos incluidas en el menú “Visualización 3D”, se pueden distinguir dos submenús, uno para visualizar los archivos completos y otro para visualizar únicamente puntos correspondientes a determinadas clases. Dentro de las opciones de visualización del archivo LiDAR completo, se pueden asignar colores en función clasificación de los puntos, por las líneas de vuelo o por intensidad de la señal.

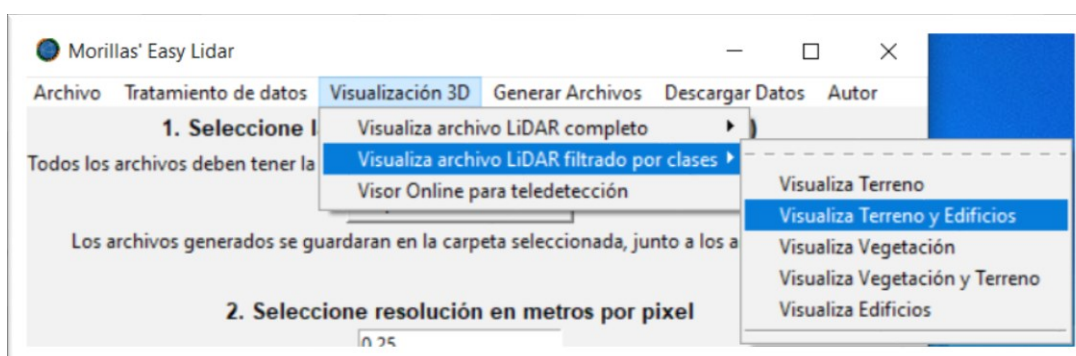


Figura 9.10. Submenú de visualización tridimensional de nube de puntos filtrados por clases.

Fuente: elaboración propia

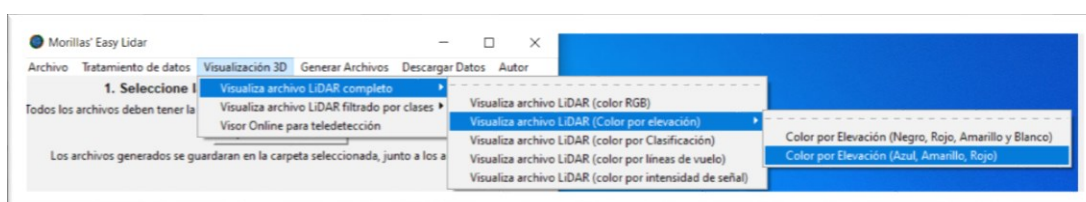


Figura 9.11. Submenú de visualización tridimensional de nube de puntos con coloración por elevación del terreno.

Fuente: elaboración propia

Dentro de las opciones de asignación de color al visualizar tridimensionalmente un archivo LiDAR completo, para la asignación de colores por alturas tenemos dos paletas de color diferentes.

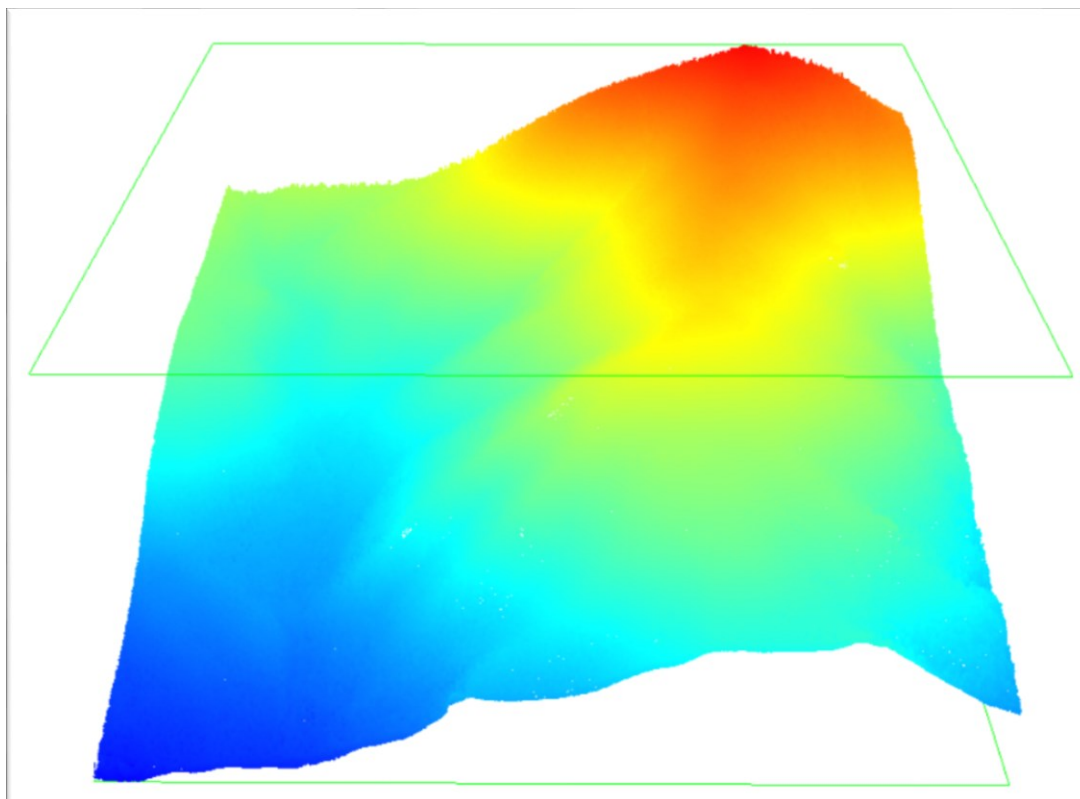


Figura 9.12. Ejemplo de visualización de nube de puntos del Valle de San Luis (Colorado, Estados Unidos) con color por altura, elaborada a partir del archivo CO_San-Luis-Valley_2011_003996 2014-08-27.laz, obtenido del (USGS).
Fuente: Elaboración propia a partir de datos de U.S. Geological Survey

En el ejemplo anterior puede comprobarse cómo el software desarrollado es capaz de trabajar eficazmente con archivos de nubes de puntos de diferente procedencia, sin limitarse a los datos de España.

A continuación, se muestran algunos ejemplos de renderizado tridimensional realizado mediante las opciones del menú “Visualización 3D” sobre datos procedentes del IGN.

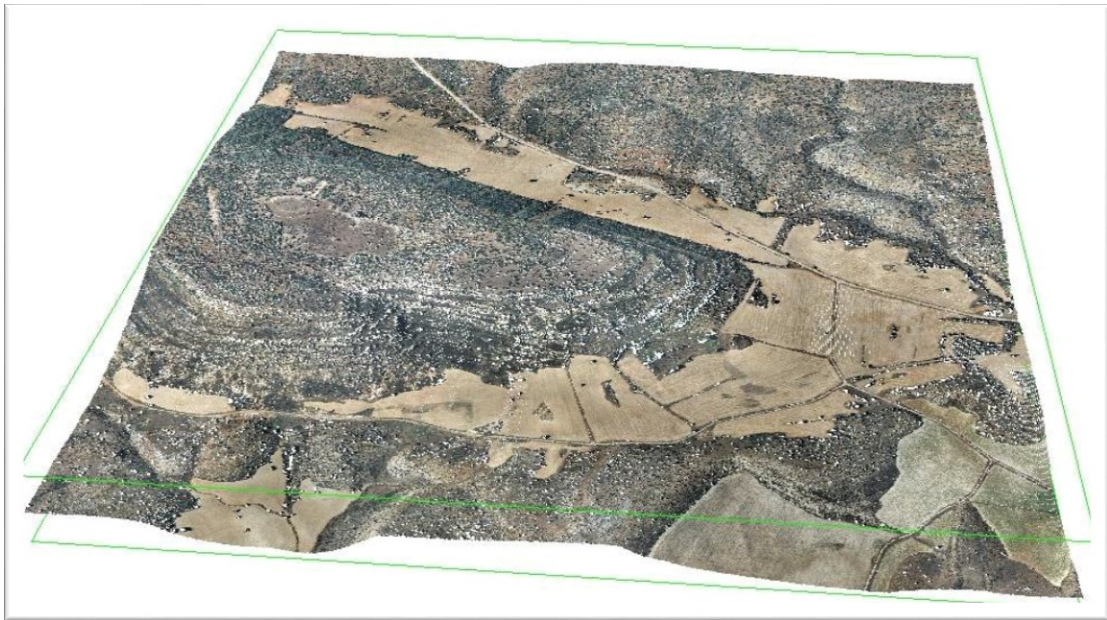


Figura 9.13. Visualización tridimensional de nube de puntos con colores RGB
Fuente: elaboración propia a partir de datos del IGN

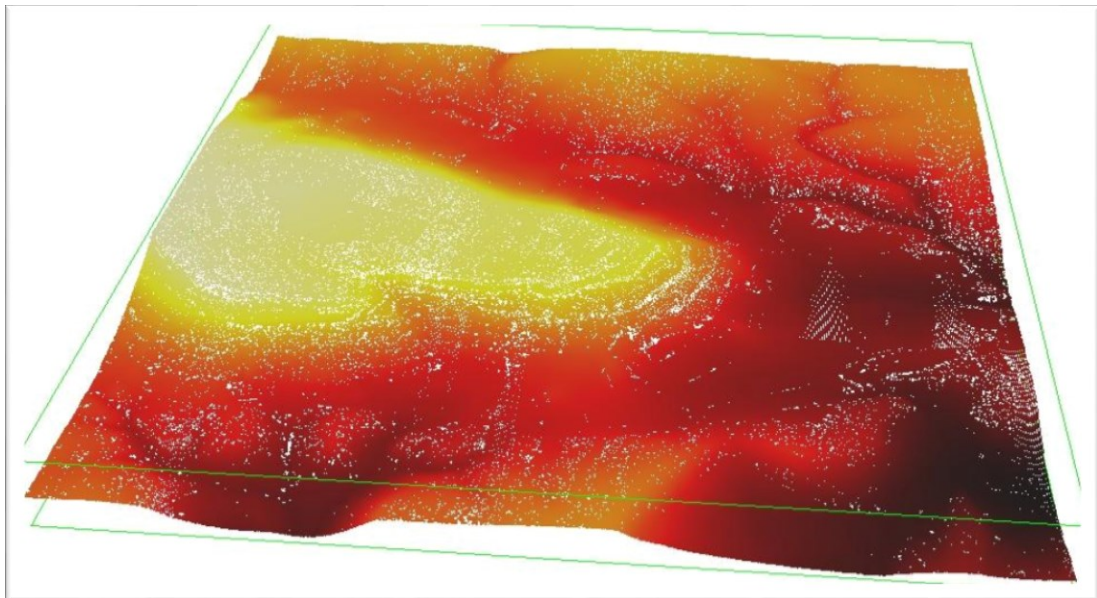


Figura 9.14. Visualización tridimensional de nube de puntos con colores por alturas (negro, rojo, amarillo y blanco)
Fuente: elaboración propia a partir de datos del IGN

El software MEL también permite la asignación de colores, en base a la intensidad con la que se recibe el retorno del láser a través sensor LiDAR.

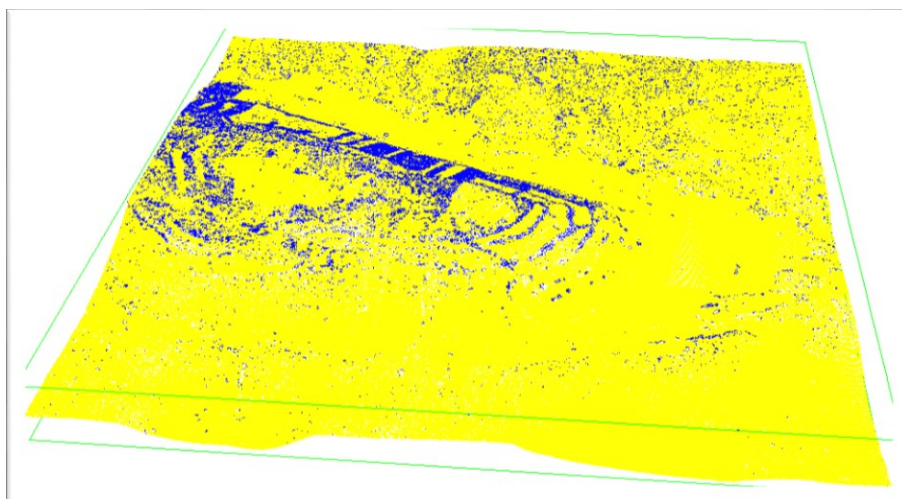


Figura 9.15. Visualización tridimensional de nube de puntos con colores por intensidad de retorno de la señal láser del LiDAR

Fuente: elaboración propia, a partir de datos del IGN

Pueden mostrarse, asimismo, los diferentes barridos realizados por los aviones del proyecto LiDAR-PNOA.

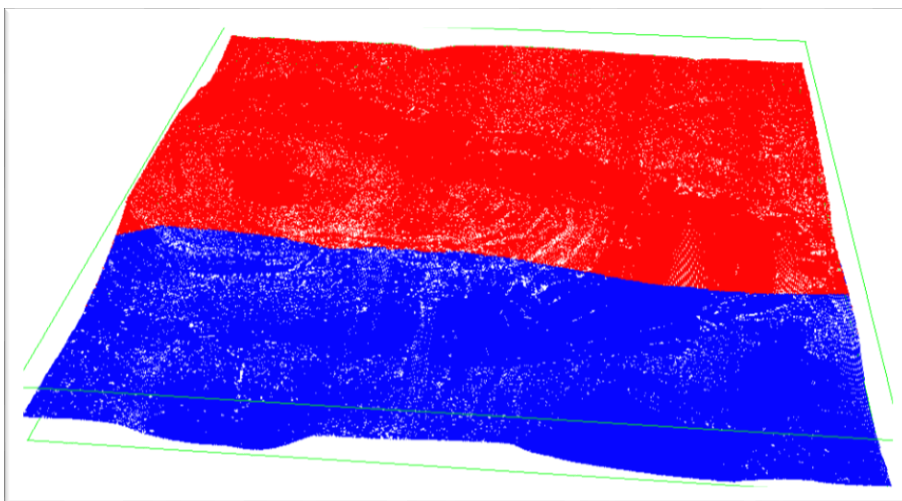


Figura 9.16. Visualización tridimensional de nube de puntos con colores por barridos aéreos realizados

Fuente: elaboración propia, a partir de datos del IGN

Una vez se ha previsualizado el archivo descargado y se han comprobado las posibles deficiencias que pudiese tener, puede procederse al tratamiento de los datos. Para facilitar esta tarea al usuario, en el software MEL se ha incluido el menú de “Tratamiento de datos”.

Dentro del menú de “Tratamiento de datos” se han creado dos submenús en función del tipo de datos a tratar, el primero de ellos es el submenú de “Nube de puntos”, el cual incluye distintas operaciones para poder unir o dividir archivos LiDAR, para realizar distintos filtrados, eliminación de datos erróneos, o duplicados, etc.

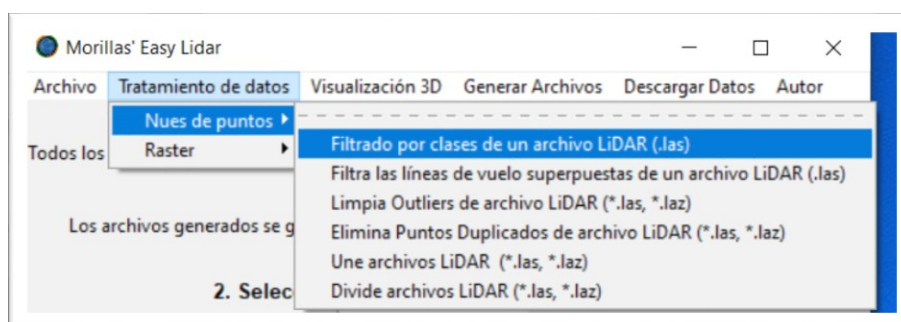


Figura 9.17. Opción de filtrado por clases de los archivos LiDAR incluida en el menú “Tratamiento de datos”

Fuente: elaboración propia

Algunas de las herramientas accesibles mediante los menús, despliegan nuevas ventanas para que el usuario pueda establecer determinados criterios a la hora de manejar los datos, éste es el caso del filtrado por clases de un archivo LiDAR.

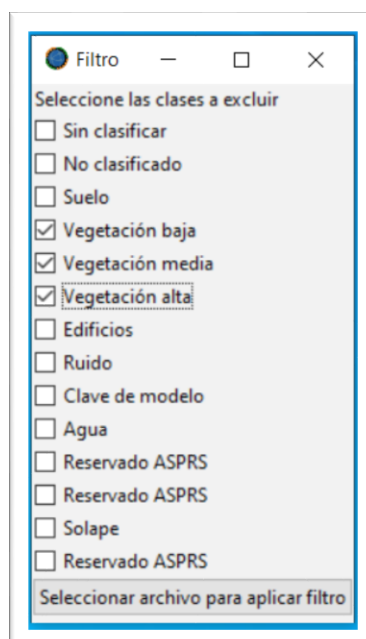


Figura 9.18. Interfaz secundaria para el filtrado de clases

Fuente: elaboración propia

El segundo submenú dentro del menú “Tratamiento de datos” es el submenú “Raster”, el cual incluye operaciones para tratar los datos ráster.

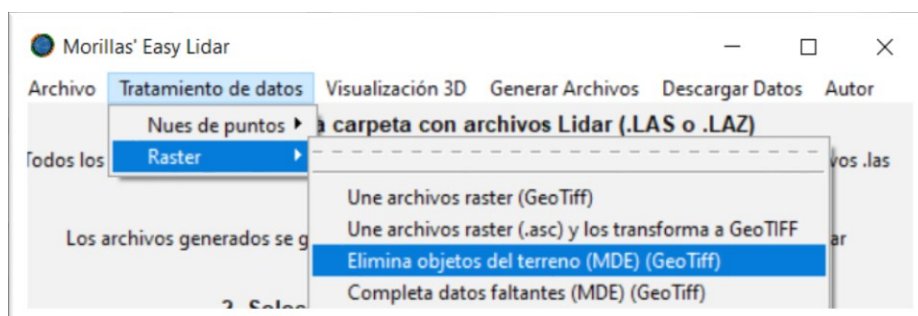


Figura 9.19. Opción de limpieza de datos mediante la eliminación de objetos del terreno incluida en el menú “Tratamiento de datos”

Fuente: elaboración propia

La fase de tratamiento de datos también incluye la generación de datos derivados, y la transformación de archivos; para estos casos se ha incluido el menú “Generar Archivos”.

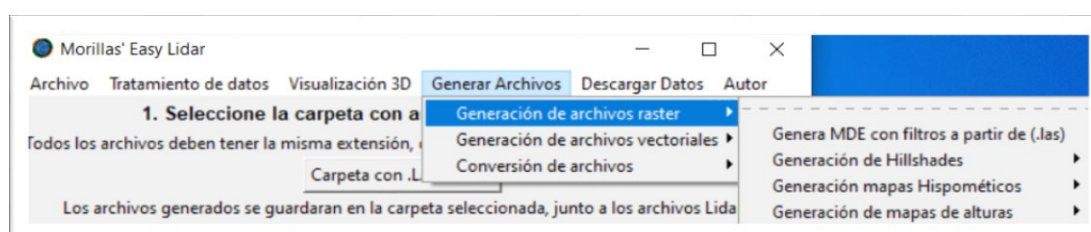


Figura 9.20. Submenú de “Generación de archivos raster”

Fuente: elaboración propia

Dentro del menú “Generar Archivos”, se han incluido tres submenús, uno para la generación de archivos ráster, otro para la generación de archivos vectoriales y el último para la conversión de archivos.

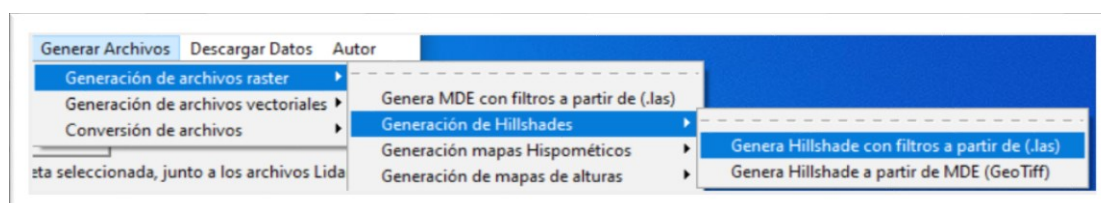


Figura 9.21. Opción para la creación de hillshades con filtros personalizados

Fuente: elaboración propia

Dentro del submenú “Generación de archivos ráster” se permite al usuario generar datos derivados tanto de ficheros ráster como de nubes de puntos.

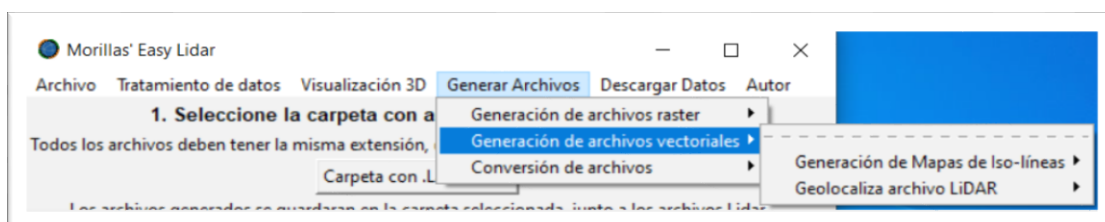


Figura 9.22. Submenú de “Generación de archivos vectoriales”

Fuente: elaboración propia

En el submenú “Generación de archivos vectoriales”, tenemos las opciones de generar mapas de Iso-líneas, tanto en formato shapefile como en formato .kml, con distintas configuraciones, y geolocalizar archivos LiDAR.

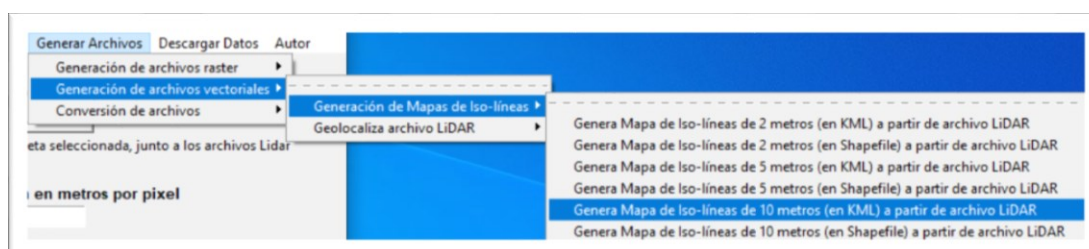


Figura 9.23. Opciones para la creación de mapas de Iso-líneas a partir de ficheros LiDAR

Fuente: elaboración propia

Para geolocalizar archivos LiDAR, se ofrece al usuario la opción de geolocalizar un único archivo LiDAR, o varios simultáneamente, incluyendo esas geolocalizaciones en archivo shapefile o kml.

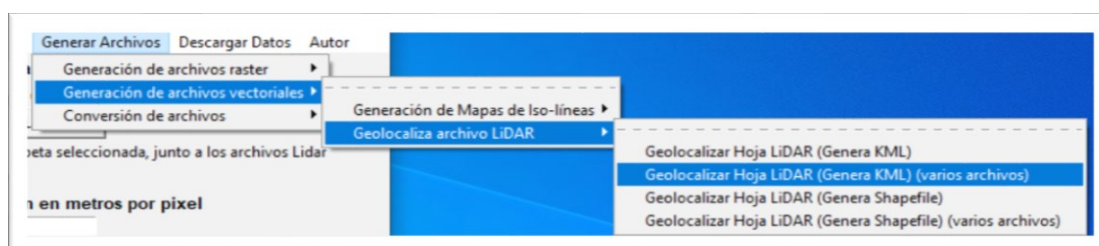


Figura 9.24. Submenú para la generación de archivos que geolocalizan la extensión de la nube de puntos

Fuente: elaboración propia

Por otra parte, para facilitar la geolocalización de los archivos LiDAR, se ha incluido la opción de generar distintos tipos de archivos (KML, WKT y Shapefile), con las geometrías correspondientes a la extensión al archivo LIDAR que indique el usuario.

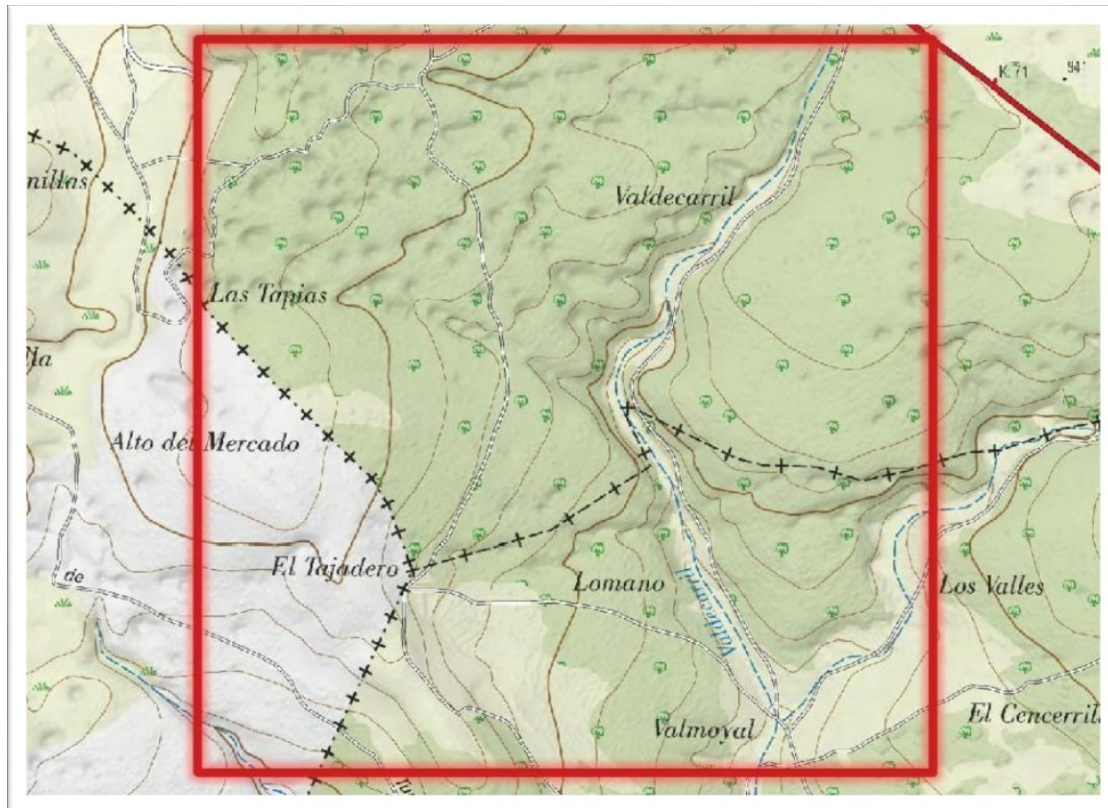


Figura 9.25. Detalle de la extensión del archivo de nube de puntos LiDAR utilizado, sobre cartografía del IGN. EPSG:25830 (ETRS89 / UTM zona 30N - Proyectado)
Fuente: elaboración propia, a partir de datos del IGN

Además de las herramientas que ya se han mencionado, el software MEL incluye en el menú “Archivo” algunas de operaciones frecuentes al trabajar con datos de nubes de puntos. Uno de los mayores problemas que se encuentran los usuarios al comenzar a trabajar con este tipo de datos es la necesidad de descomprimir los archivos .LAZ (como los que ofrece el IGN) a formato .LAS; debido a esto se han incluido las opciones de descompresión de archivos tanto de forma individual como masiva.

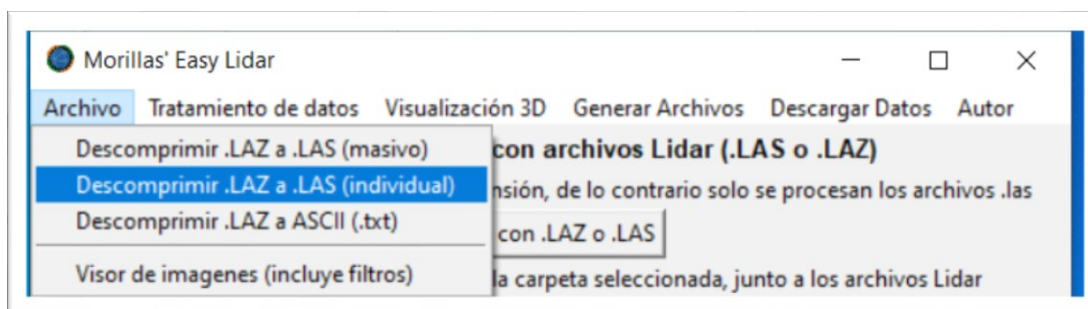


Figura 9.26. Opciones para la descompresión de archivos LiDAR incluidas en el menú “Archivo”

Fuente: elaboración propia

Por último cabe mencionar que, para poder extraer conocimiento de los datos, es necesario visualizarlos; los datos que genera la aplicación pueden ser integrados sin problemas en otros sistemas de información geográfica como QGIS, pero si se quiere hacer una previsualización rápida de los archivos ráster generados o aplicarles determinados filtros, se ha incluido también en el menú “Archivo” un visor/editor de imágenes ligero basado en IrfanView 64 (software libre para su uso no comercial), el cual permite manipular archivos pesados sin problemas.

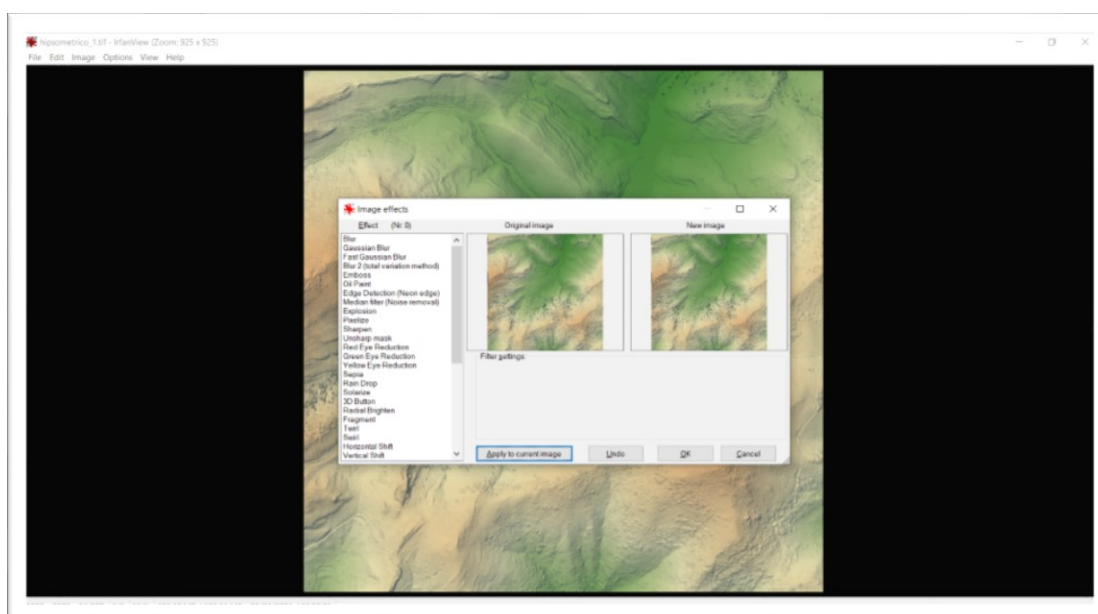


Figura 9.27. Ejemplo de carga de un mapa hipsométrico en el visor/editor de archivos
Fuente: elaboración propia, a partir de datos del IGN

9.2. Generación automatizada de imágenes individuales a partir de datos LiDAR

Para la elaboración de las imágenes LiDAR que se muestran en los casos de uso del capítulo 10, en primer lugar se obtienen los datos del Centro de descargas del CNIG (IGN) (<http://centrodedescargas.cnig.es/CentroDescargas/index.jsp>), mediante la búsqueda por coordenadas. Las coordenadas de los lugares escogidos para elaborar el catálogo de zonas de interés, se basa en el listado de lugares de interés geológico del IGME (<http://info.igme.es/ielig/ListaLIGs.aspx>), algunos de los cuales están incluidos como Global Geosites dentro del Proyecto Global Geosites, coordinado por la Unión Internacional de Ciencias Geológicas y la Unesco para la protección del patrimonio geológico internacional. Varios de estos sitios, por su mayor relevancia, son los utilizados para los casos de uso recogidos en este documento.



Figura 9.28. Centro de descargas del IGN (búsqueda por coordenadas UTM)
Fuente: elaboración propia, a partir de datos del IGN

Una vez encontrado el archivo LiDAR correspondiente, se procede a su descarga. Teniendo en cuenta que no todas las zonas de España son ofrecidas en datos de la segunda cobertura LiDAR, en ocasiones este proceso se realiza con los archivos generados en la primera cobertura, dando preferencia a los de la segunda, en caso de estar disponibles, dada la mayor resolución que ofrecen.

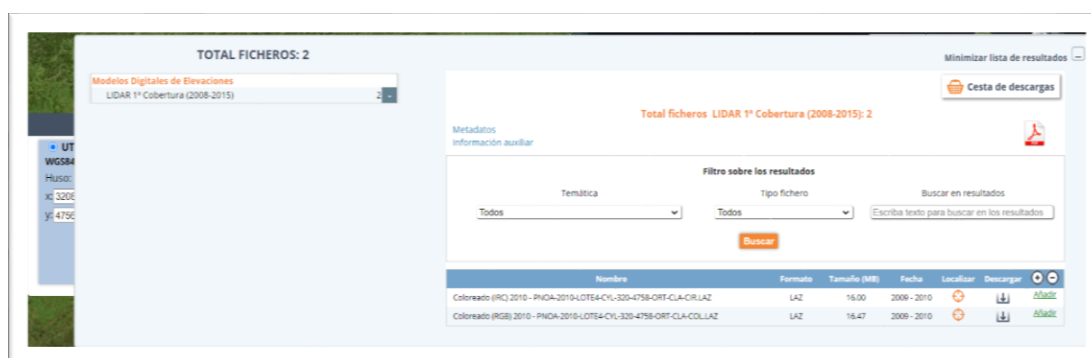


Figura 9.29. Selección de archivos de nube de puntos desde el centro de descargas del IGN
Fuente: elaboración propia, a partir de datos del IGN

Cada hoja ofrecida por el IGN, en ambas coberturas, está disponible tanto con coloración RGB como IRC, siendo la primera opción la utilizada en esta investigación, ya que ofrece los colores del espectro visible elaborados con las ortoimágenes del proyecto PNOA.

En este proceso se realiza la descarga individual de los archivos, a diferencia del proceso de descarga masivo descrito en el epígrafe 9.3 de este capítulo. Una vez descargado el archivo objeto de estudio, debe procesarse, ya que la nube de puntos LiDAR, viene comprimida con extensión .laz; para ello se utiliza el software de creación propia MEL, el cual permite tanto la descompresión, como el procesamiento y la generación de imágenes derivadas.

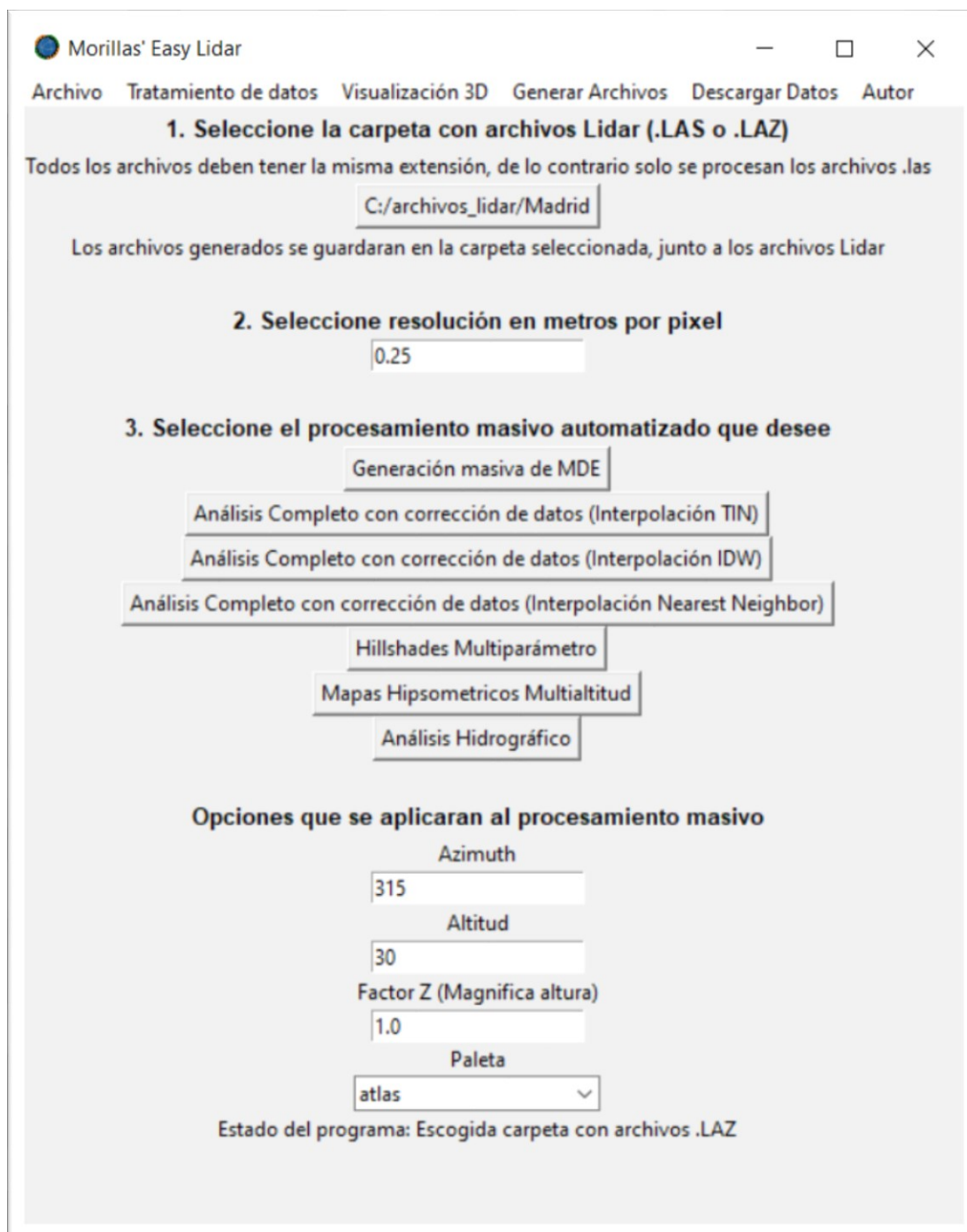


Figura 9.30. Interfaz del software de tratamiento de datos LiDAR (MEL), después de haber indicado la fuente de archivos de nube de puntos
Fuente: elaboración propia

En segundo lugar, se indican los parámetros que van a definir el hillshade y el mapa hipsométrico, siendo los más adecuados, en la mayoría de los casos, los siguientes: azimuth 315°; altitud 30° y factor z = 1 (los cuales se han definido

por defecto); éste último se mantiene en 1, para no modificar la relación de altura con las dimensiones de la hoja, la paleta definida por defecto es “atlas”, las distintas paletas de colores aplicables a los mapas hipsométricos generados son las utilizadas más habitualmente en la visualización de datos. En caso de que el hillshade o el mapa hipsométrico no fuesen correctos, pueden analizarse distintas combinaciones de parámetros de forma masiva, aplicando las opciones “Hillshades Multiparámetro” o “Mapas Hipsométricos Multialtitud”, además de poder magnificar la altura con el Factor Z, en caso de querer resaltar alguna estructura. Una vez definidos los parámetros, se procede a realizar un “Análisis Completo con corrección de datos”, gracias al cual se corrigen irregularidades y errores existentes en los datos ofrecidos por el IGN, además de filtrar los puntos extrayendo únicamente los correspondientes al terreno; gracias a esto se pueden apreciar estructuras cubiertas por la vegetación.

A continuación, se muestran todos los archivos generados, algunos de los cuales son archivos intermedios generados durante la limpieza y el enriquecimiento de los datos.

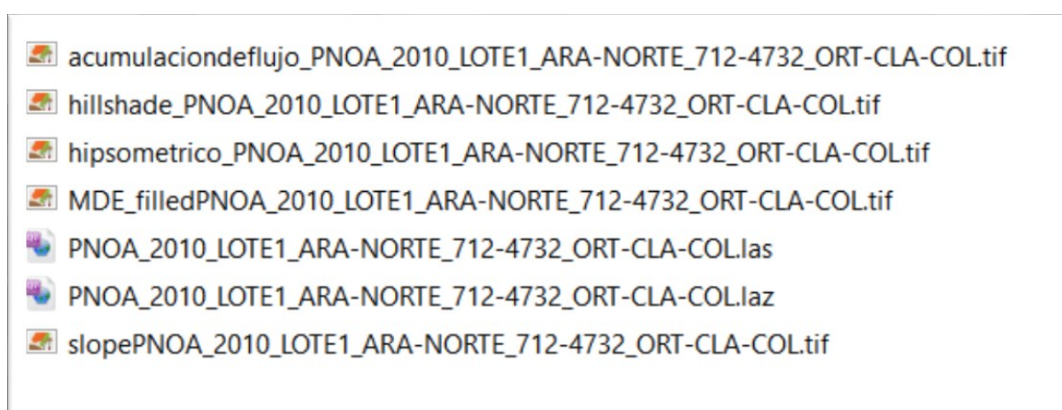


Figura 9.31. Vista de los archivos generados por el software MEL
Fuente: elaboración propia

A continuación, se muestran algunos de los GeoTiff generados de forma automática con la configuración predefinida.

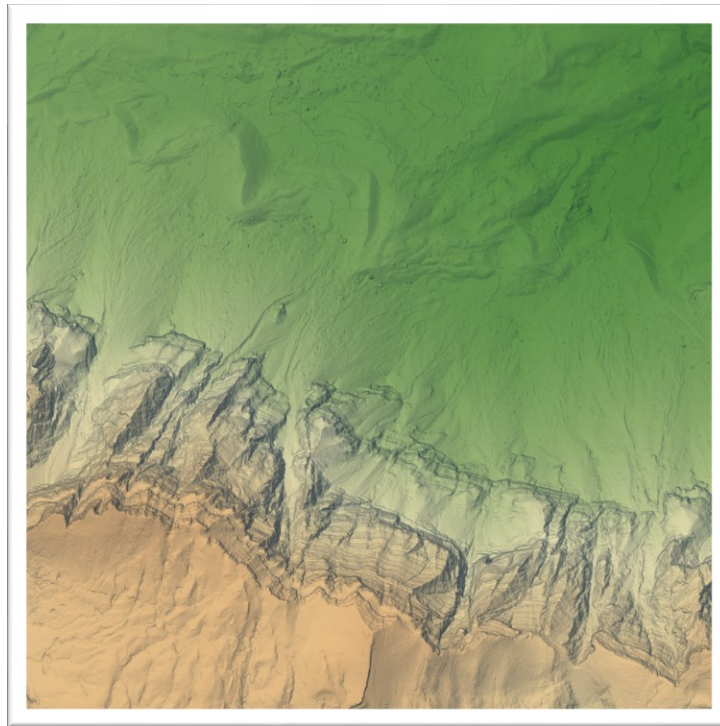


Figura 9.32. Mapa hipsométrico (paleta de color “atlas”) generado mediante el software MEL
Fuente: elaboración propia, a partir de datos del IGN.

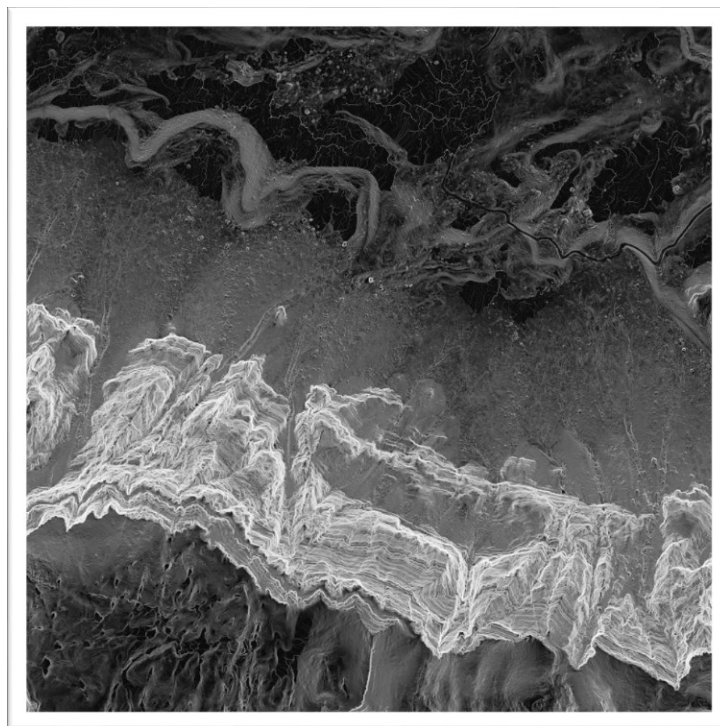


Figura 9.33. Mapa de pendientes generado mediante el software MEL
Fuente: elaboración propia, a partir de datos del IGN.

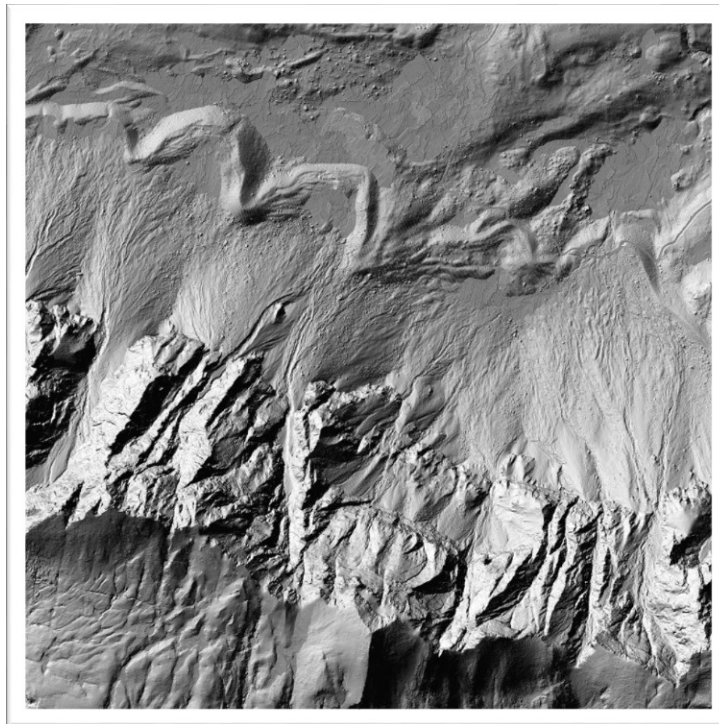


Figura 9.34. Hillshade generado mediante el software MEL
Fuente: elaboración propia, a partir de datos del IGN.



Figura 9.35. MDE generado mediante el software MEL
Fuente: elaboración propia, a partir de datos del IGN.

Para generar mapas hipsométricos con una paleta de colores diferente, únicamente es necesario cambiar la paleta seleccionada en el menú desplegable situado en la parte inferior de la interfaz.

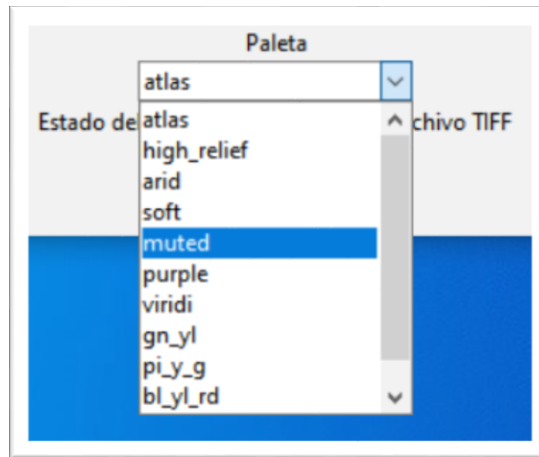


Figura 9.36. Selección de la paleta de colores
Fuente: elaboración propia

En el siguiente ejemplo se muestra el resultado obtenido con la paleta de colores “muted”.

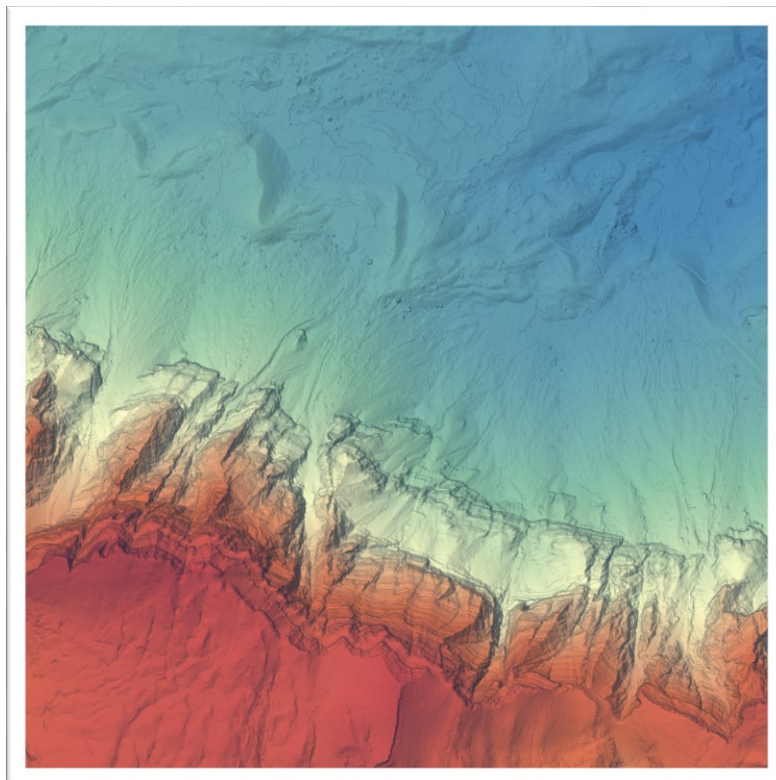


Figura 9.37. Mapa hipsométrico (paleta de color “muted”) generado mediante el software MEL
Fuente: elaboración propia, a partir de datos del IGN

Para combinar los distintos archivos generados o establecer colores en función de la altura (con otras configuraciones diferentes a las obtenidas en el mapa hipsométrico generado automáticamente), se utiliza el software QGIS, alimentado con los archivos generados por MEL, ya que también genera modelos digitales de elevaciones.

Para mostrar los colores en función de su altitud, se aplica el modo de renderizado de pseudocolor monobanda al modelo digital de elevaciones.

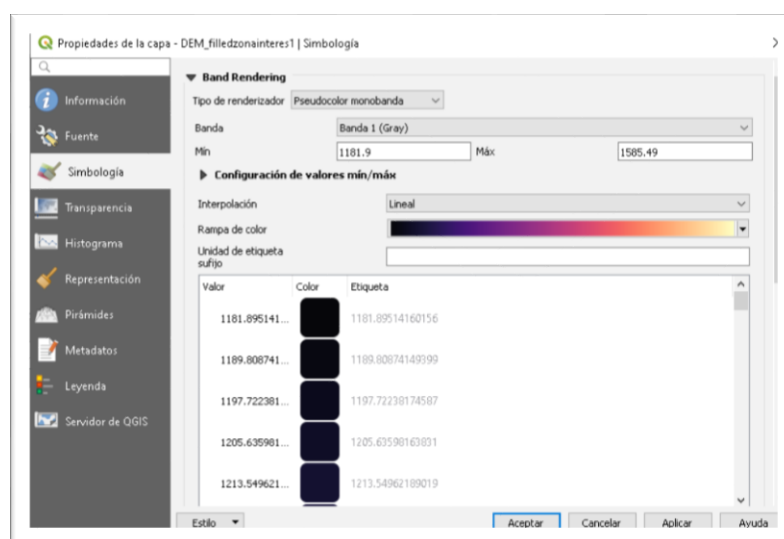


Figura 9.38. Propiedades de renderizado del modelo digital de elevaciones generado con el software de elaboración propia, cargado en QGIS

Fuente: elaboración propia

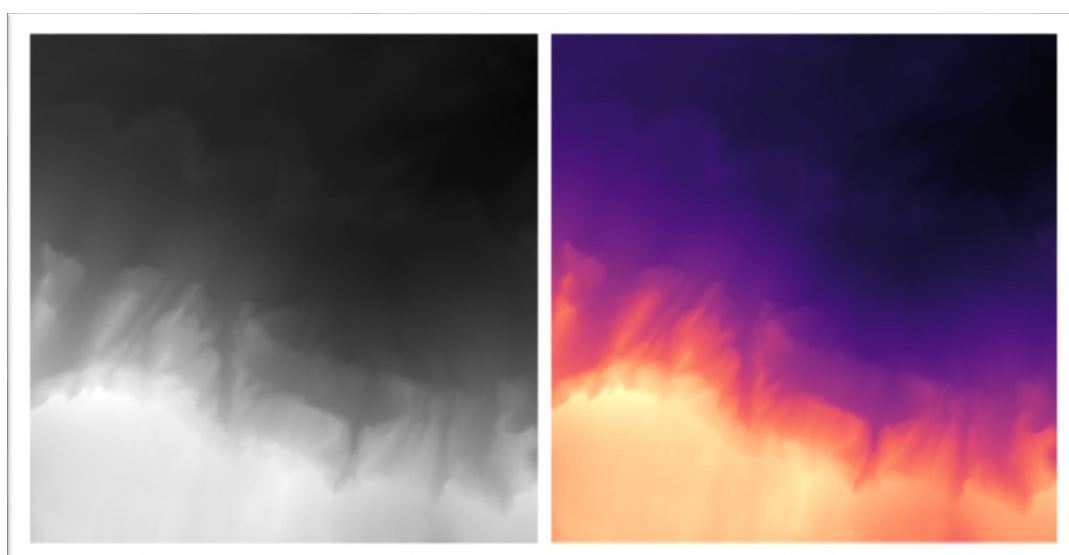


Figura 9.39. MDE original y MDE con pseudocolor monobanda.

Fuente: elaboración propia, a partir de datos del IGN

Finalmente, se modifica su transparencia y se combina con el hillshade, produciendo el resultado mostrado en la siguiente imagen.

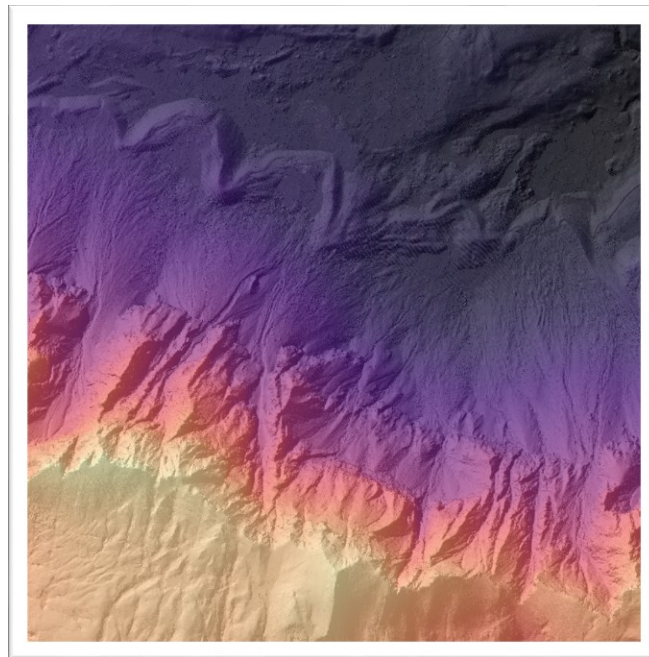


Figura 9.40. Hillshade generado con el software MEL, coloreado en base a las alturas contenidas en el MDE

Fuente: elaboración propia, a partir de datos del IGN

Para dar una coloración diferente a las rampas de color predefinidas por QGIS, se pueden definir los colores manualmente.

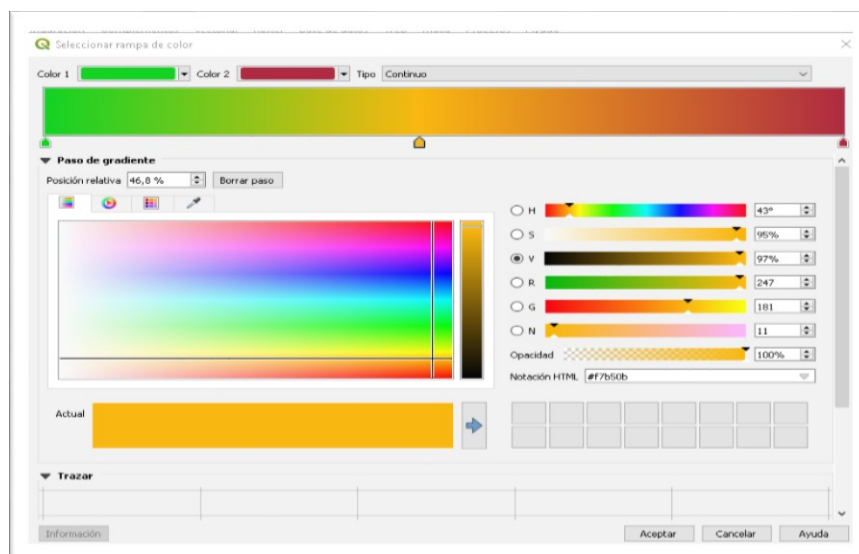


Figura 9.41. Selección de la rampa de color en QGIS

Fuente: elaboración propia

Otro de los archivos que genera el análisis con MEL es el flujo acumulado, el cual representa el peso acumulado de todas las celdas que fluyen hacia cada celda con pendiente descendente en el ráster de salida.

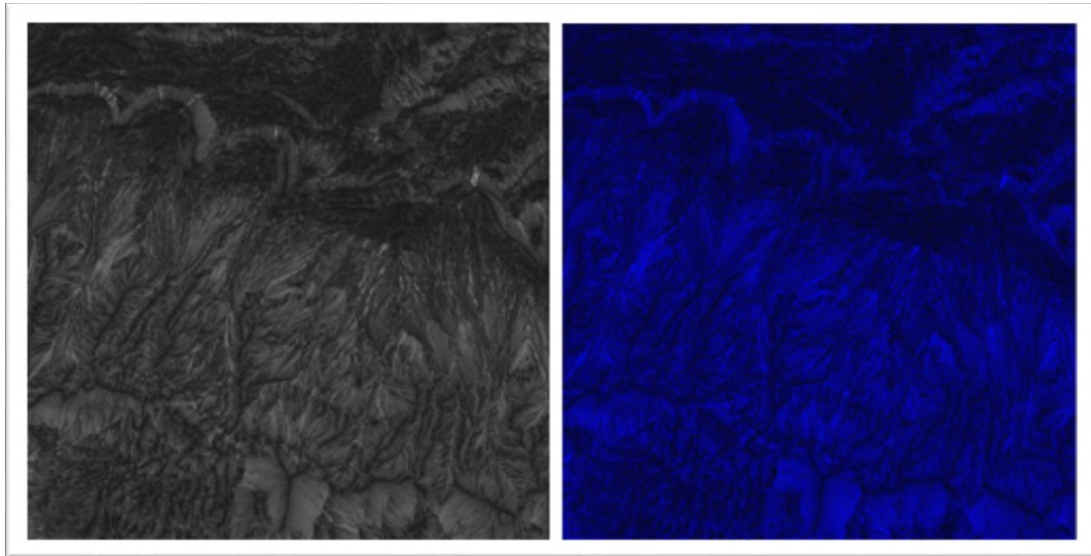


Figura 9. 42. Mapa de acumulación de flujo generado con el software MEL en gris monobanda y color multibanda
Fuente: elaboración propia, a partir de datos del IGN

Modificando el color del renderizado del mapa de acumulación de flujo, se puede dar coloración al mismo; en este caso, usando el color de multibanda y asociando la escala de grises al color azul.

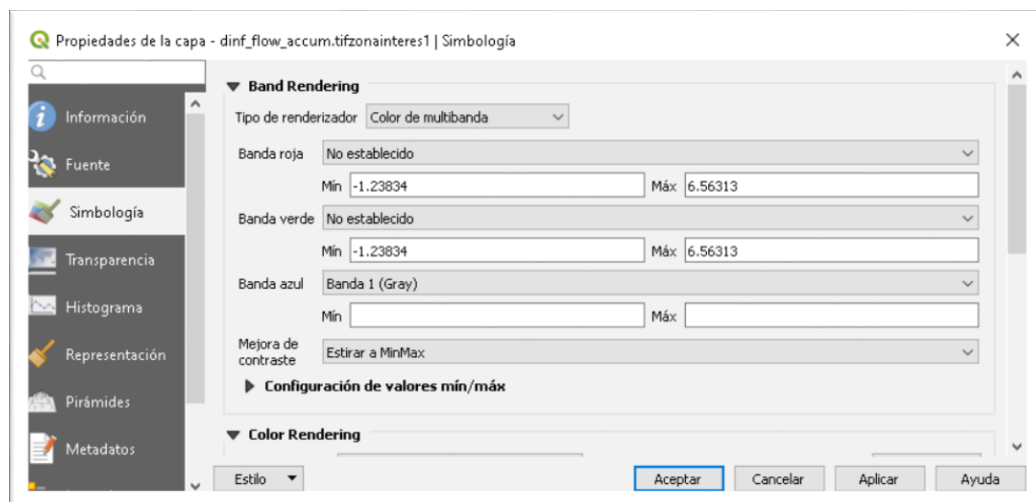


Figura 9.43. Propiedades de renderizado del mapa de acumulación de flujo en QGIS
Fuente: elaboración propia

Una vez hecho esto, se pueden combinar todas las capas, lo cual puede ser de gran utilidad en estudios hidrográficos.

Al ráster de acumulación de flujo, también se le puede aplicar un pseudocolor monobanda.

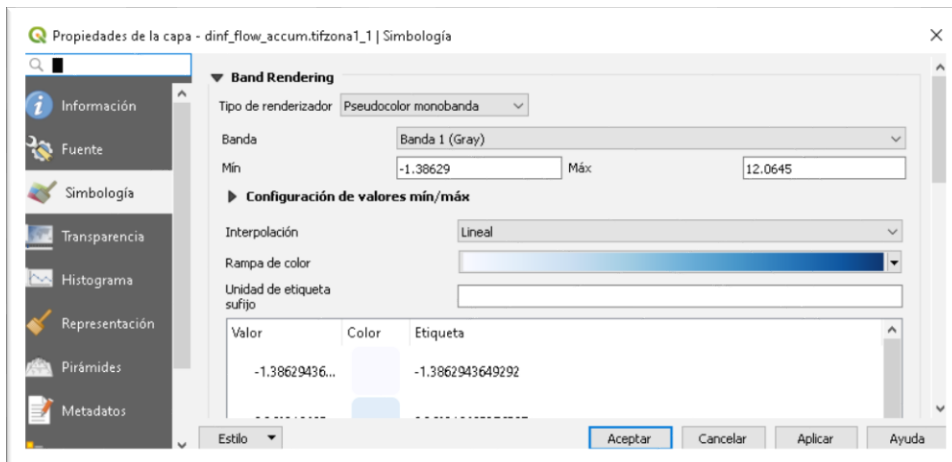


Figura 9.44. Propiedades de renderizado del mapa de acumulación de flujo, para el establecimiento de pseudocolor monobanda en QGIS

Fuente: elaboración propia

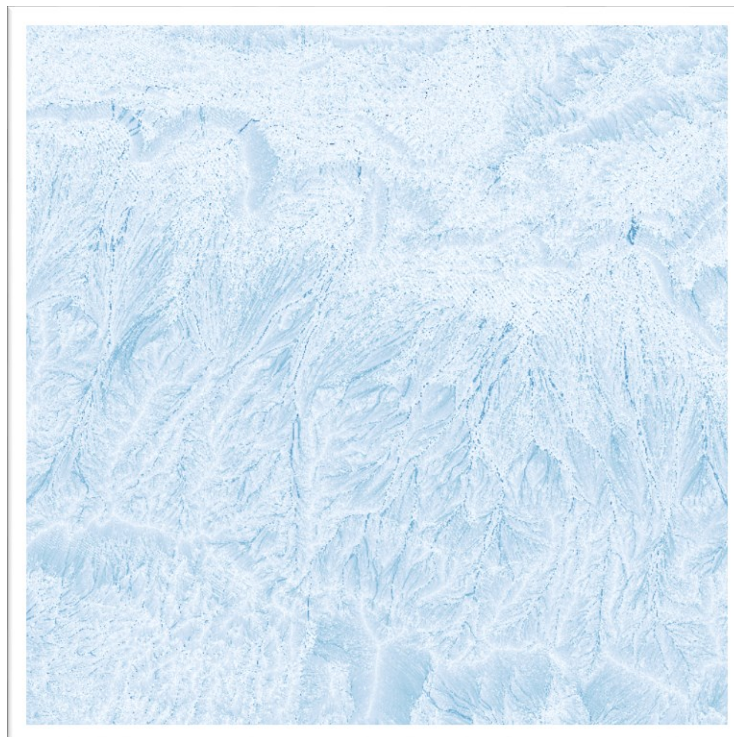


Figura 9.45. Mapa de acumulación de flujo, con pseudocolor monobanda (25cm/px)

Fuente: elaboración propia, a partir de datos del IGN

De esta forma, se ha realizado el análisis del relieve con múltiples combinaciones, para identificar estructuras tanto geomorfológicas como estructuras de origen antrópico, de las cuales, por motivos de espacio, solamente se muestran algunas.

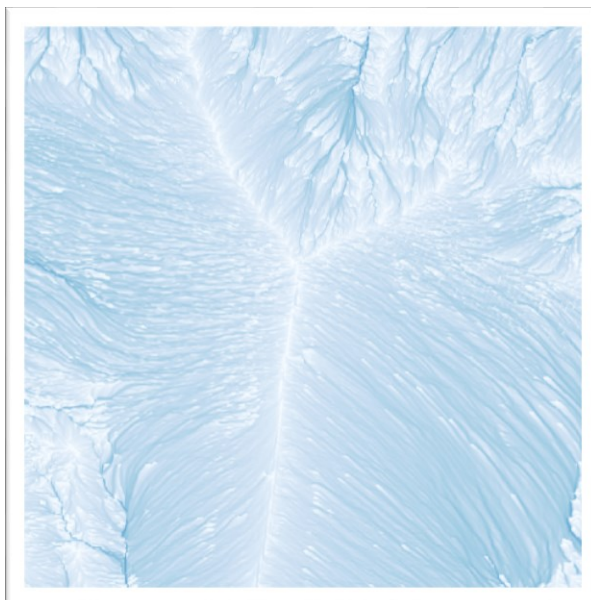


Figura 9.46. Detalle del mapa de acumulación de flujo, con pseudocolor monobanda (25cm/px)
Fuente: elaboración propia, a partir de datos del IGN

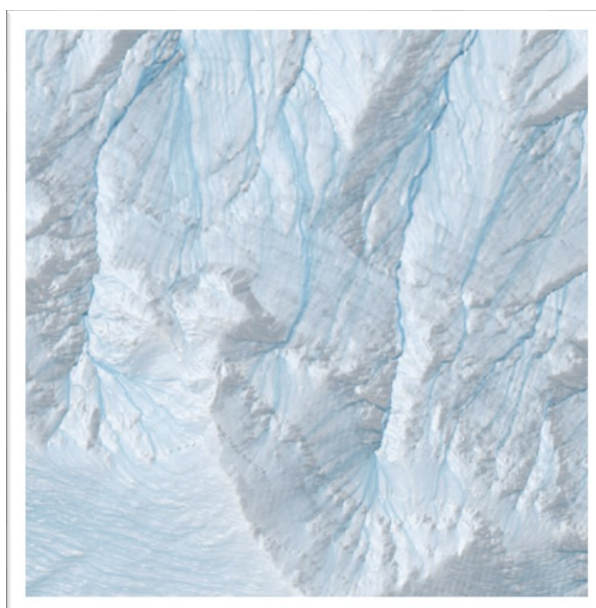


Figura 9.47. Detalle de la combinación del hillshade con el mapa de acumulación de flujo, ambos generados con el software de creación propia (MEL)
Fuente: elaboración propia, a partir de datos del IGN

Al igual que ocurre con los mapas de acumulación de flujo, a los mapas de pendiente también se le puede asignar pseudocolor monobanda, siguiendo el mismo procedimiento descrito anteriormente, tal y como se muestra en la siguiente figura.

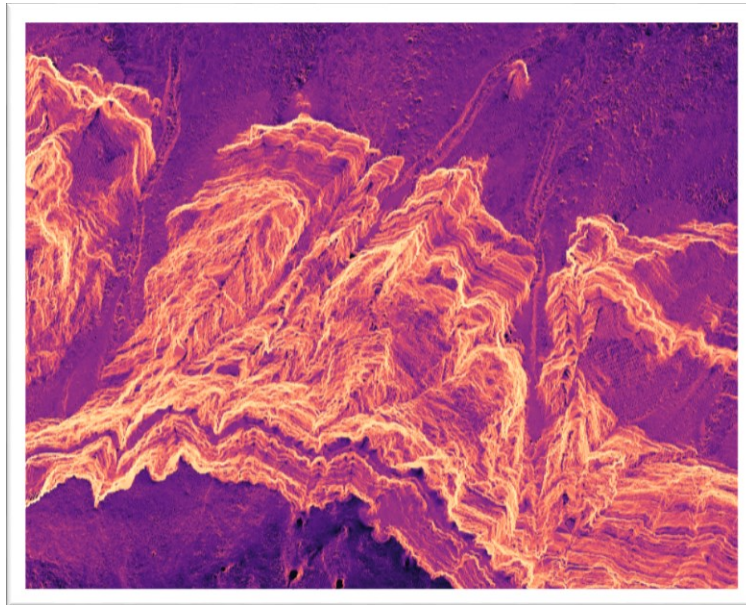


Figura 9.48. Detalle del mapa de pendientes con pseudocolor monobanda.
Fuente: elaboración propia, a partir de datos del IGN

9.3. Elaboración masiva de imágenes a partir de datos LiDAR, para el análisis exploratorio y la detección de estructuras

Para poder realizar el reconocimiento visual del relieve, deben generarse las imágenes ráster, derivadas de los datos contenidos en las nubes de puntos ofrecidas por el IGN, como hillshades, mapas de pendientes, acumulación de flujo, etc. El software MEL se ha diseñado para cubrir este objetivo, permitiendo la ingesta masiva de datos mediante la selección por carpeta, lo que facilita el trabajo. Es necesario indicar que, previamente, debe realizarse la descarga de archivos.

La descarga masiva de ficheros LiDAR desde el centro de descargas del Centro Nacional de Información Geográfica (<http://centrodedescargas.cnig.es>), se lleva a cabo a través de distintas opciones que se ofrecen de selección de los archivos a descargar.



Figura 9.49. Opciones de búsqueda de archivos en el centro de descargas del IGN

Fuente: elaboración propia

Para la realización del análisis del relieve de un área determinada, la forma de búsqueda de ficheros que resulta más conveniente es mediante la búsqueda por hojas, ya que, de esta forma, podemos seleccionar todos los archivos LiDAR que contienen la información sobre el territorio definido por una hoja del MTN50 y así sistematizar el trabajo.



Figura 9.50. Descarga de archivos del IGN mediante la búsqueda por hojas del MTN50
Fuente: elaboración propia, a partir de datos del IGN

Para ayudarnos a localizar la hoja de interés, puede activarse la opción ver mapa con numeración del MTN50; de esta forma podremos ver los números de hoja sobre el mapa y así seleccionar exactamente la hoja a analizar. Esta fase puede obviarse si se dispone de la numeración de la hoja de la que se quieren obtener los datos del LiDAR.

Debido a que la metodología propuesta consiste en la descarga masiva de archivos LiDAR del IGN, correspondientes a una hoja del MTN50, para su posterior tratamiento, filtrado y transformación en un modelo digital de elevaciones con mayor resolución que el MDT05 o el MDT02, suministrados por el IGN, se opta por proseguir con la descarga masiva mediante la búsqueda a través de la hoja MTN50.

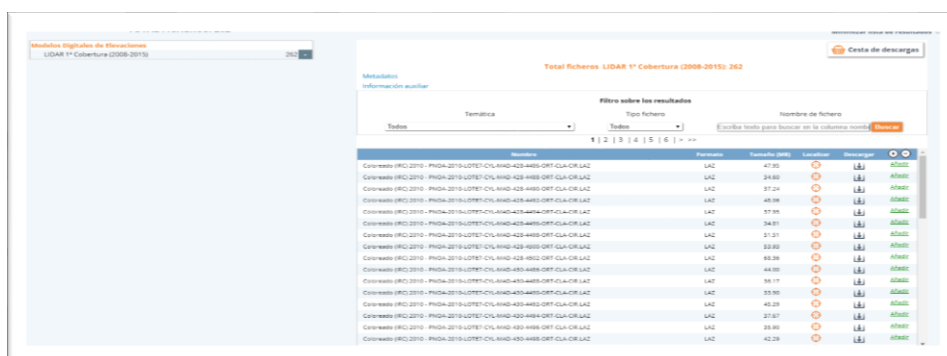
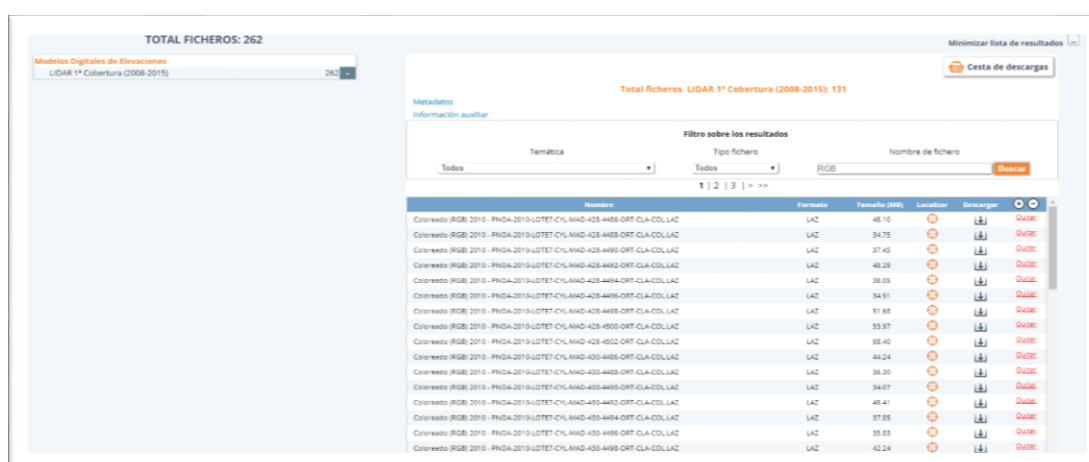


Figura 9.51. Vista del panel de descargas de los archivos LiDAR correspondientes a la hoja MTN50 seleccionada

Fuente: elaboración propia, a partir de datos del IGN

Una vez realizada la búsqueda de los archivos LiDAR de interés para la investigación, puede comprobarse que tenemos disponibles nubes de puntos coloreadas (IRC) y nubes de puntos coloreadas (RGB); considerando que la revisión que se va a realizar es principalmente el análisis regional del relieve, se opta por las que están coloreadas con los colores del espectro visible (RGB), para automatizar la descarga de, únicamente, los archivos RGB.

Para poder realizar esto, debe escribirse RGB dentro del cuadro de búsqueda “Nombre de Fichero” y hacer clic en Buscar; tras realizar esta acción puede comprobarse cómo ha disminuido el número de ficheros, ya que sólo están disponibles los coloreados por RGB.



TOTAL FICHEROS: 262

Modelos Digitales de Elevaciones

LiDAR 1ª Cobertura (2008-2015)

Minimizar lista de resultados

Cesta de descargas

Total ficheros LiDAR 1ª Cobertura (2008-2015): 131

Metadatos

Información auxiliar

Filtro sobre los resultados

Temática: Todos

Tipo fichero: Todos

Nombre de fichero: RGB

1 2 3 > >>

Nombre	Formato	Tamaño (MB)	Localizar	Descargar
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	48.10		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	34.75		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	37.40		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	46.28		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	38.05		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	34.91		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	51.68		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	53.97		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	68.40		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	44.24		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	36.30		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	34.07		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	45.41		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	37.85		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	35.83		
Colorado (RGB) 2010 - PHDA-2010-LOTET-CYL-MAD-425-4488-ORT-CLA-COL-LAZ	LAZ	42.24		

Figura 9.52. Filtrado e inclusión de los archivos correspondientes a la hoja MTN50 seleccionada en la cesta de descarga para la descarga masiva de archivos
Fuente: elaboración propia a partir de datos del IGN

A fin de no tener que seleccionar cada uno de los archivos de forma independiente, podemos hacer clic en el icono (+, Seleccionar todos); esta selección incluye todos los archivos, tanto de la paginación actual como de las siguientes. En el caso del ejemplo que podemos ver en la imagen se han seleccionado a la vez los 131 archivos.

Para proceder a la descarga de todos ellos se debe hacer clic en cesta de descargas (esquina superior derecha de la Figura 9.53).

Hecho esto se presenta la pantalla que se muestra a continuación en la Figura 9.54.



Figura 9.53. Vista de los archivos incluidos en la cesta de descarga para la descarga masiva
Fuente: elaboración propia, a partir de datos del IGN

Desde esta interfaz puede procederse al inicio de la descarga, pudiendo incluirse más productos a descargar, pueden eliminarse productos, o puede vaciarse la cesta. En este caso la selección que corresponde es Iniciar Descarga. Hecho esto, la página del IGN nos muestra un aviso legal en el que se especifica la licencia de uso, la cual es compatible con Creative Commons.



Figura 9.54. Aviso legal de licencia de uso de los archivos descargados desde el IGN
Fuente: elaboración propia, a partir de datos del IGN

La orden ministerial a la que hace referencia está disponible a través del siguiente enlace: <https://www.boe.es/boe/dias/2015/12/26/pdfs/BOE-A-2015-14129.pdf>

Antes de proceder a la descarga, el IGN solicita la participación en una encuesta de uso de la información geográfica, la cual puede obviarse haciendo clic en No Enviar y continuar la descarga. Hecho esto, la página de descargas del IGN nos pide una última confirmación de los ficheros a descargar.

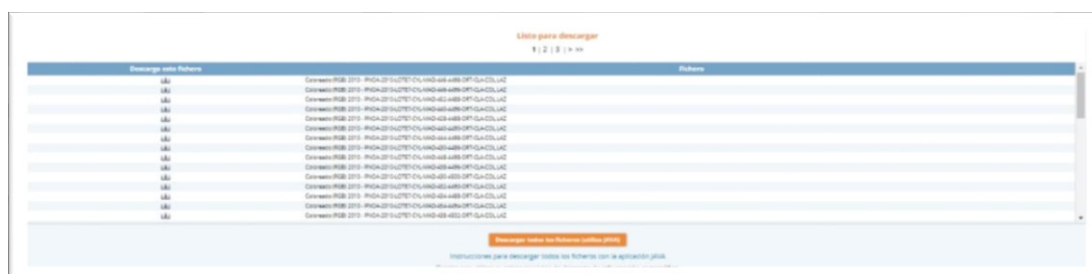


Figura 9.55. Confirmación de los ficheros a descargar en la descarga masiva
Fuente: elaboración propia, a partir de datos del IGN

Una vez comienza la descarga, el navegador que se esté utilizando puede dar un aviso de seguridad, debido a que el procedimiento de descarga masiva de ficheros desde el IGN se realiza a través de una aplicación Java.

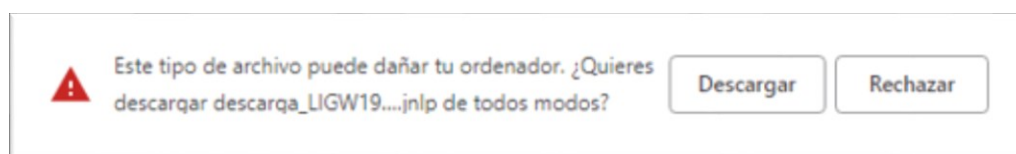


Figura 9.56. Aviso de seguridad al descargar la aplicación de descarga masiva del IGN
Fuente: elaboración propia

El programa Java es seguro, por lo que se selecciona Descargar y se continúa con la descarga. Una vez se ha descargado el programa Java, éste debe ejecutarse.



Figura 9.57. Confirmación de ejecución del programa Java de descarga masiva del IGN
Fuente: elaboración propia

La interfaz del programa Java para la descarga de archivos del IGN es la que se muestra a continuación.

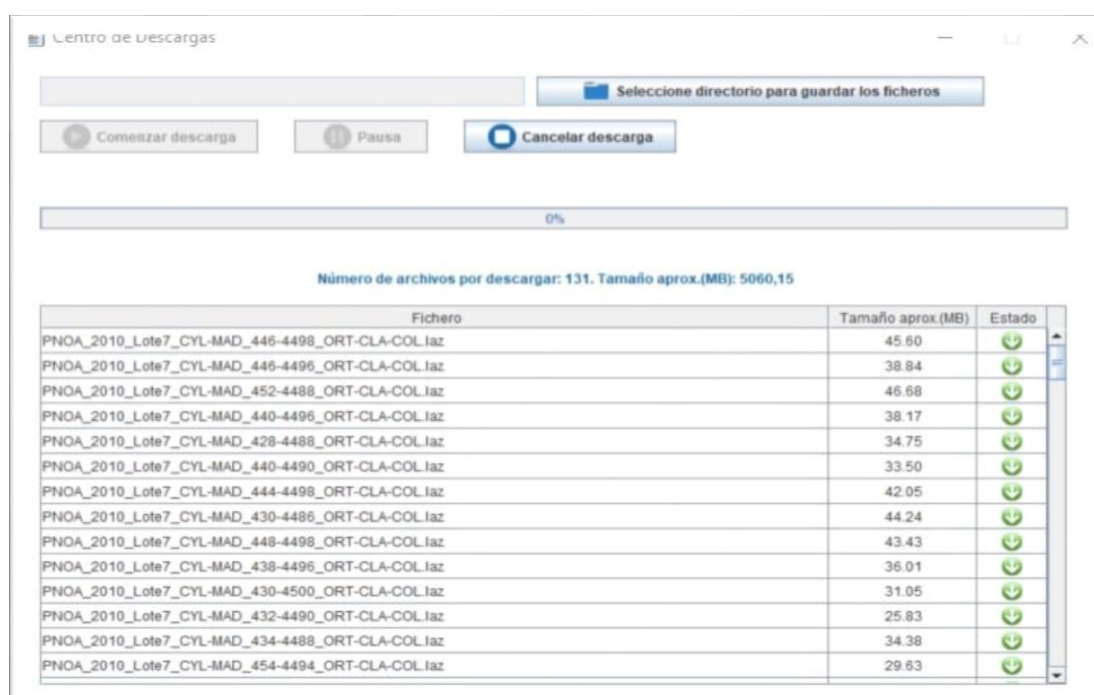


Figura 9.58. Interfaz del programa de descarga masiva de archivos del IGN
Fuente: elaboración propia

Desde esta interfaz podemos iniciar la descarga, pausarla, cancelarla y seleccionar el directorio donde se van a guardar los ficheros descargados. Para tal fin se ha creado una carpeta dentro del directorio raíz para almacenar los ficheros descargados; se ha escogido esta ubicación para reducir la ruta de los archivos y evitar posibles problemas en el proceso de ingesta de datos.

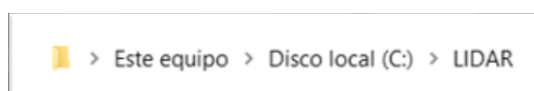


Figura 9.59. Dirección local de la carpeta donde se descargan los archivos masivamente
Fuente: elaboración propia

Dentro de esta ruta, se han creado, a su vez, un conjunto de subcarpetas con el nombre de cada una de las hojas MTN50, de las cuales se van a descargar los archivos LiDAR correspondientes, por lo que se selecciona para la descarga la carpeta con el nombre de la hoja MTN50 correspondiente. Una vez se ha seleccionado la carpeta en la que se van a guardar los ficheros se procede a comenzar con la descarga, tal y como se muestra en la siguiente imagen.

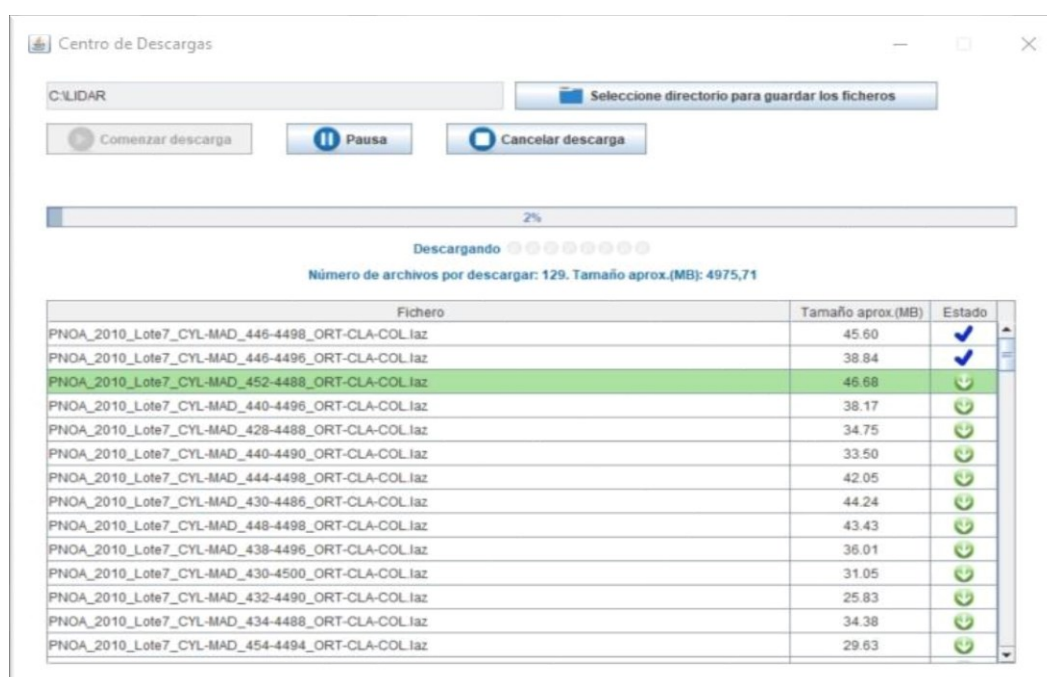


Figura 9.60. Proceso de descarga masiva de archivos mediante el software Java del IGN
Fuente: elaboración propia

Hay que tener siempre presente que el tiempo necesario para descargar los archivos va a depender, entre otros factores, de la velocidad de nuestra conexión a la red.

Podemos comprobar que los archivos LiDAR que ofrece el IGN vienen servidos con la extensión .laz, es decir han sido comprimidos mediante la librería de compresión desarrollada por Martin Isenburg, LASzip, la cual ha sido integrada en el software desarrollado para la descompresión de los ficheros servidos por el IGN.

Una vez finalizada la descarga de los archivos que van a ser utilizados, se procede a la selección de la carpeta donde han sido alojados, a través del software MEL.



Figura 9.61. Selección de carpeta con los archivos LiDAR mediante el software de elaboración propia para el procesamiento de nubes de puntos MEL

Fuente: elaboración propia

Seleccionada la carpeta con los archivos que van a ser procesados, se selecciona la opción de “Análisis completo con corrección de datos”. De esta forma comienza el proceso masivo y automatizado de generación de los distintos archivos que se van a utilizar en el estudio del relieve y la identificación de estructuras antrópicas, en base a los parámetros definidos por el usuario en los cuadros de texto. El filtrado de los datos se realiza de forma automática, en base a la clasificación automática de los mismos realizada en el proyecto PNOA, conservando únicamente los correspondientes al terreno (se elimina la vegetación, el agua, los edificios, etc.). Los archivos generados por este proceso son imágenes ráster georreferenciadas en formato GeoTiff, entre las que se incluyen los modelos digitales de elevaciones generados mediante interpolación

TIN e IDW, en la cual se basan el resto de las imágenes, ya que ofrece mayor precisión. A partir del modelo TIN se generan otros MDE en los que se corrigen errores en la toma de datos con los sensores LiDAR, y se completan los huecos provocados por densidades de puntos menores en determinadas zonas. Asimismo, partiendo de los MDE corregidos se generan los ráster de hillshades, mapas de pendientes y acumulación de flujo.

Si se quiere mejorar la visibilidad de un elemento en particular, puede, además, realizarse la generación de hillshades multiparámetro, en la que automáticamente se generan hillshades con distintas configuraciones de parámetros, que nos permite comprobar cuál es la más adecuada para visualizar las estructuras a estudiar. Esto es fundamental, ya que los parámetros de altitud y azimuth son cruciales a la hora de poder identificar visualmente distintas estructuras; un ejemplo de esta funcionalidad puede observarse en el anexo II, en el cual se muestra una pequeña parte de las combinaciones que genera de forma automatizada el software aquí desarrollado (MEL).

Una vez generados los hillshades con las combinaciones adecuadas de parámetros, se procede a darles color según la altura, mediante QGIS, tal y como se ha descrito en el epígrafe anterior.

Además de la coloración según alturas, para el estudio regional de la geomorfología pueden integrarse otras capas con los MDE, para obtener MDT que incluyan la litografía del terreno, las fallas, los plegamientos, etc. Para obtener estas capas se recurre a los servicios WMS ofrecidos por el IGME.

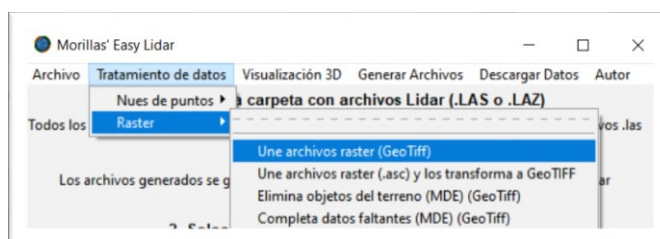


Figura 9.62. Opción de unión de archivos ráster incluida en el menú “Tratamiento de datos”
Fuente: elaboración propia

Para unir los distintos ráster generados mediante el software MEL, puede utilizarse la herramienta incluida en el software (Figura 9.63), o recurrir a QGIS, para visualizar los datos mientras se realiza el proceso. El procedimiento seguido para la unión de las diferentes capas ráster generadas consiste en cargar en QGIS las capas del área a estudiar, tal y como se muestra en la Figura 9.64, con los datos generados para la isla de El Hierro:

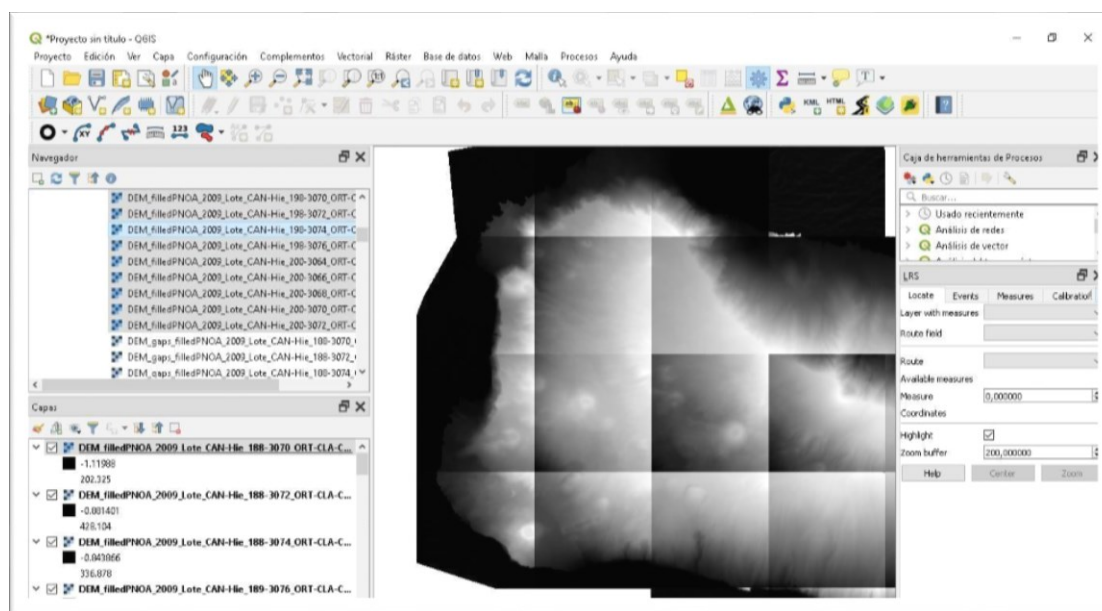


Figura 9.63. Imagen de los MDE de El Hierro unidos mediante QGIS
Fuente: elaboración propia, a partir de datos del IGN

En este ejemplo, los ráster generados a partir de las nubes de puntos LiDAR, tienen una resolución de 0,5 metros/píxel, la cual es 10 veces superior a la de los MDT ofrecidos por el IGN y permite, por lo tanto, un estudio más detallado del relieve. El sistema de referencia con el que se ha trabajado en este ejemplo es REGCAN95. Para cambiar el sistema de referencia del proyecto generado para este ejemplo y adaptarlo al adecuado para las Islas Canarias, se hace mediante las propiedades del proyecto mostradas en la Figura 9.66.

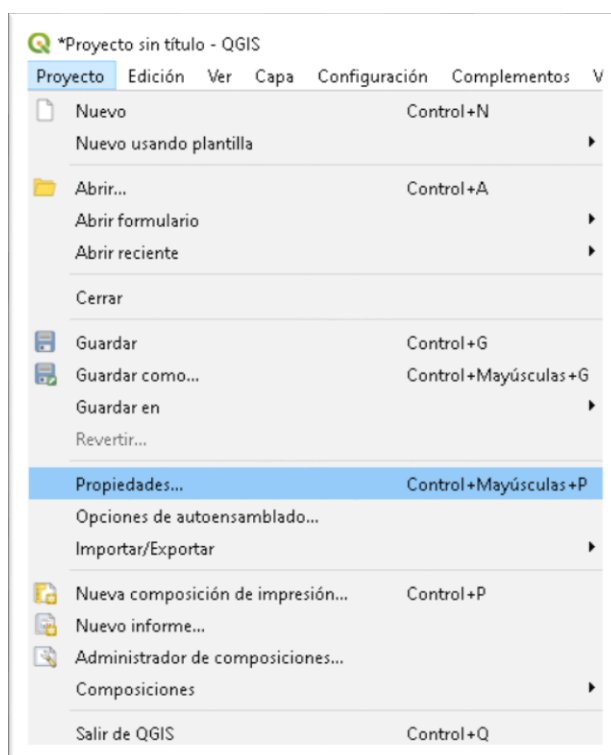


Figura 9.64. Imagen del menú de QGIS para cambiar las propiedades del proyecto
Fuente: elaboración propia

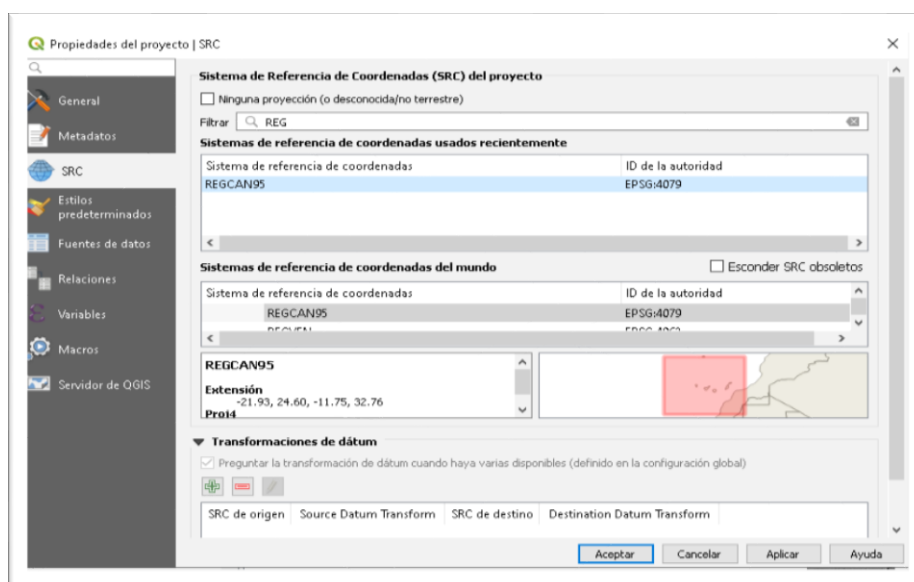


Figura 9.65. Imagen de la selección de un SRC (REGCAN95, para los datos de las Islas Canarias), mediante QGIS
Fuente: elaboración propia

Para cambiar el sistema de referencia de las capas del proyecto (si es necesario), se accede mediante el menú de Capa, seleccionando la opción "Establecer SRC de la Capa(s)".

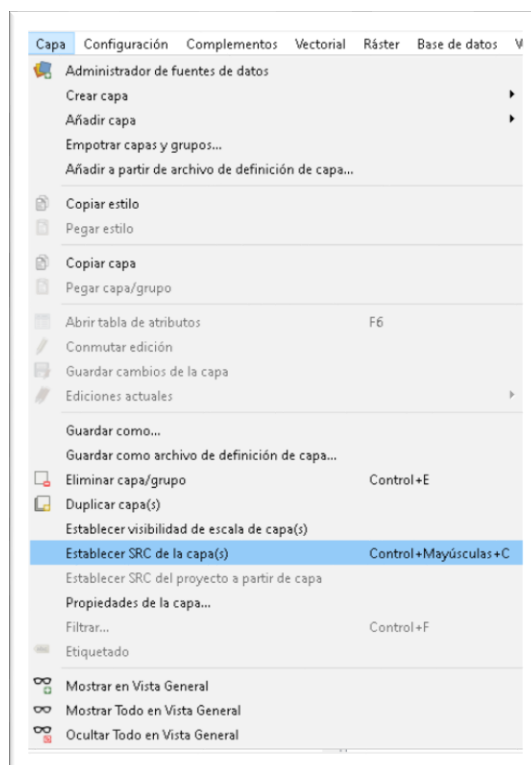


Figura 9.66. Imagen del menú de QGIS, para cambiar el SRC de las capas
Fuente: elaboración propia

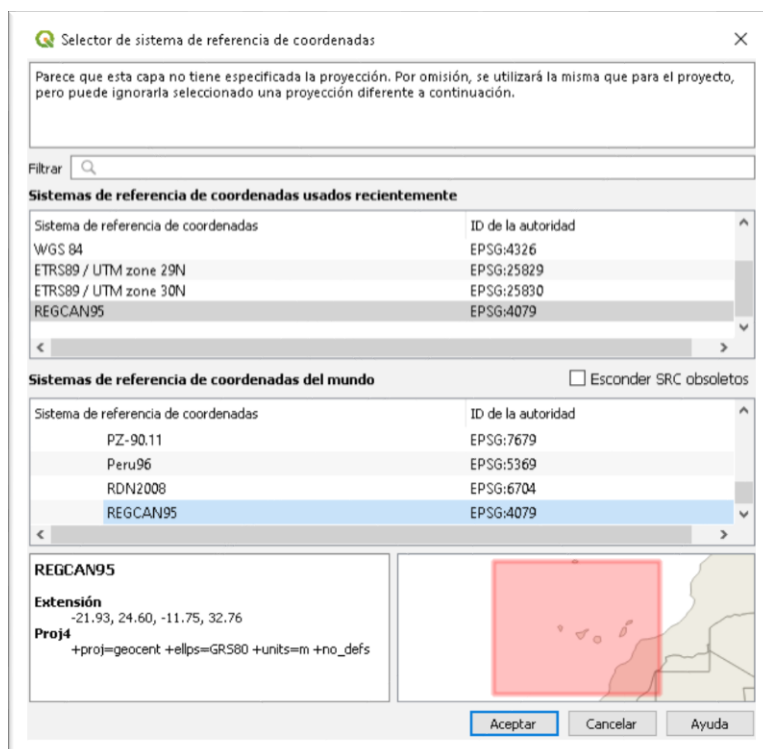


Figura 9.67. Selector de sistema de referencia de coordenadas en QGIS
Fuente: elaboración propia

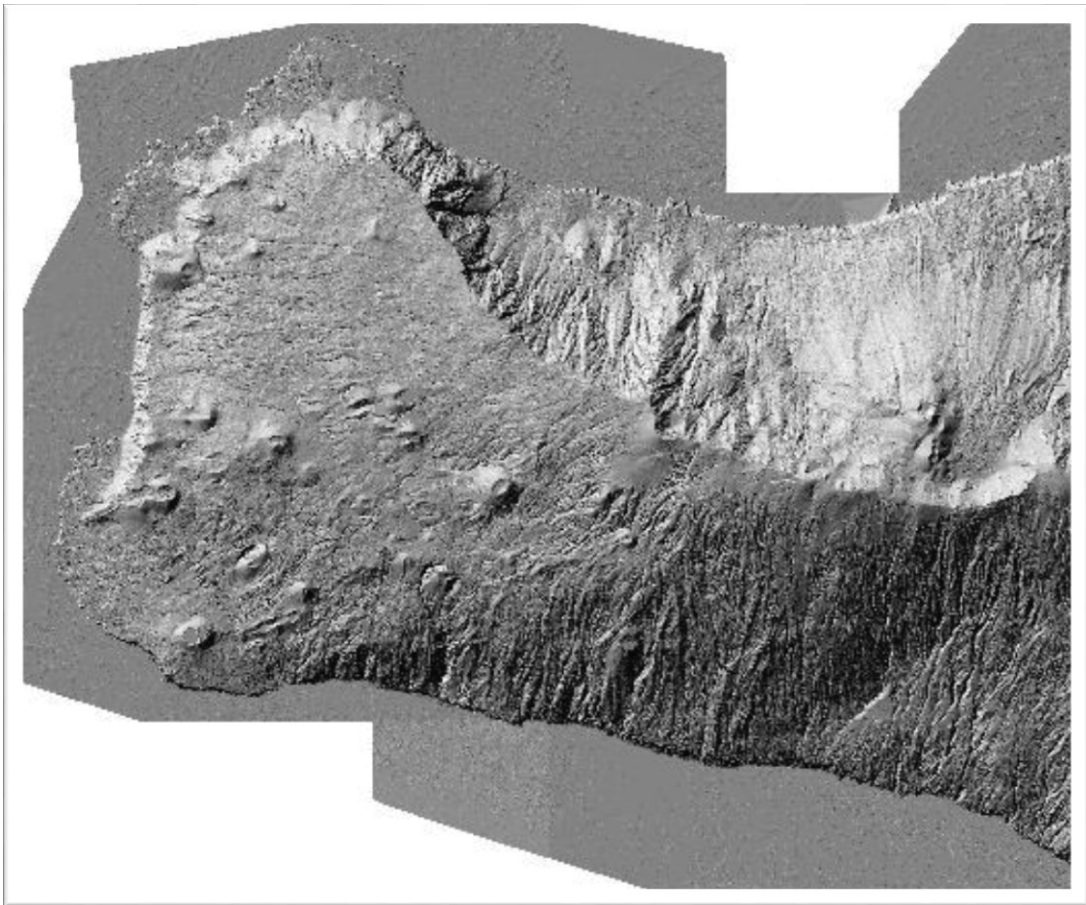


Figura 9.68. Unión de hillshades generados con MEL
Fuente: elaboración propia, a partir de datos del IGN

Una vez cargadas las capas con las que se va a trabajar, y establecido el sistema de referencia de coordenadas adecuado, se procede a la combinación de estas para el estudio del relieve.

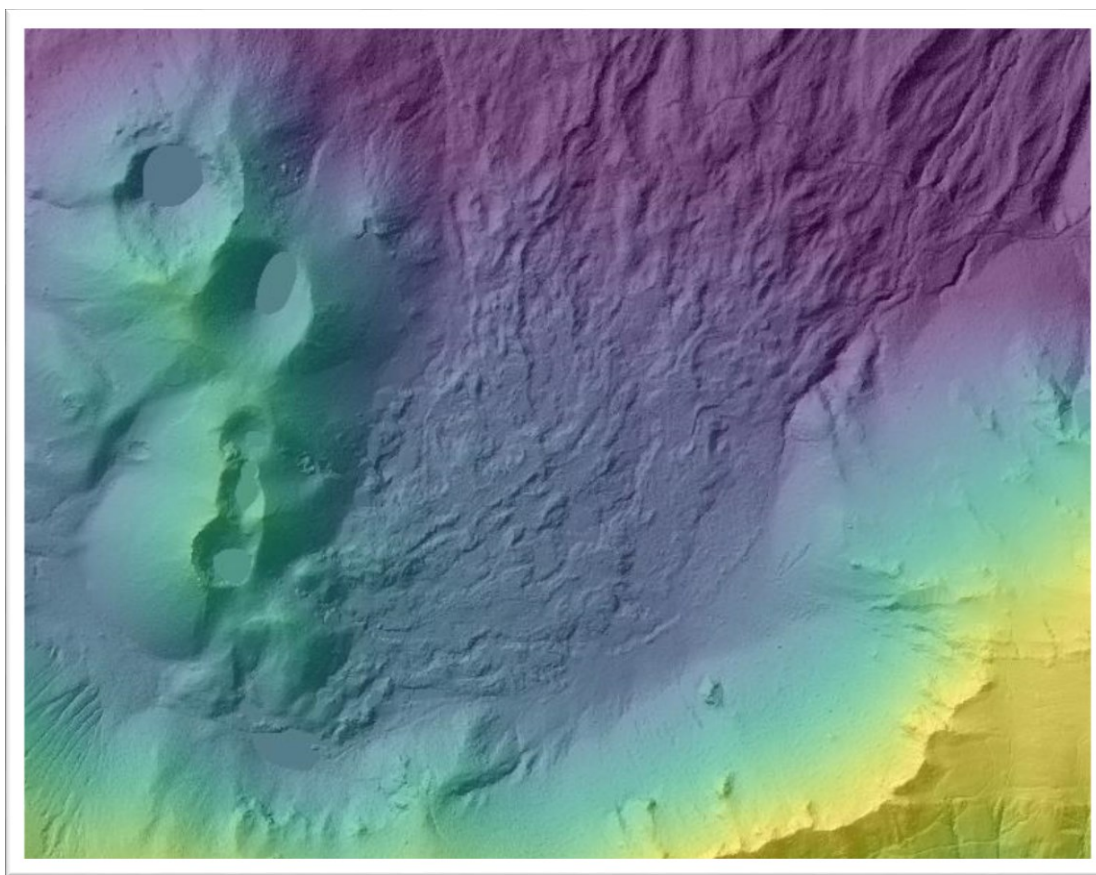


Figura 9.69. Detalle del hillshade coloreado según alturas, mediante la combinación del hillshade y el MDT generados a partir de un archivo de nube de puntos LiDAR del IGN, con una resolución obtenida de 25 cm por pixel
Fuente: elaboración propia, a partir de datos del IGN

9.4. Análisis del relieve mediante realidad virtual y tratamiento dinámico de los datos sin necesidad de descarga

Por otra parte, para la revisión de relieves de mayor extensión, de una forma más rápida, se ha creado un GIS online, que permite el estudio del relieve a nivel mundial. A este software, se le ha incluido la opción de su uso mediante Realidad Virtual, creando una vista estereoscópica de los modelos digitales de elevaciones. Con la finalidad de hacer accesible la herramienta al mayor número posible de personas, se ha alojado públicamente en la siguiente dirección. <http://cartotecnica.com/teledeteccion/>.

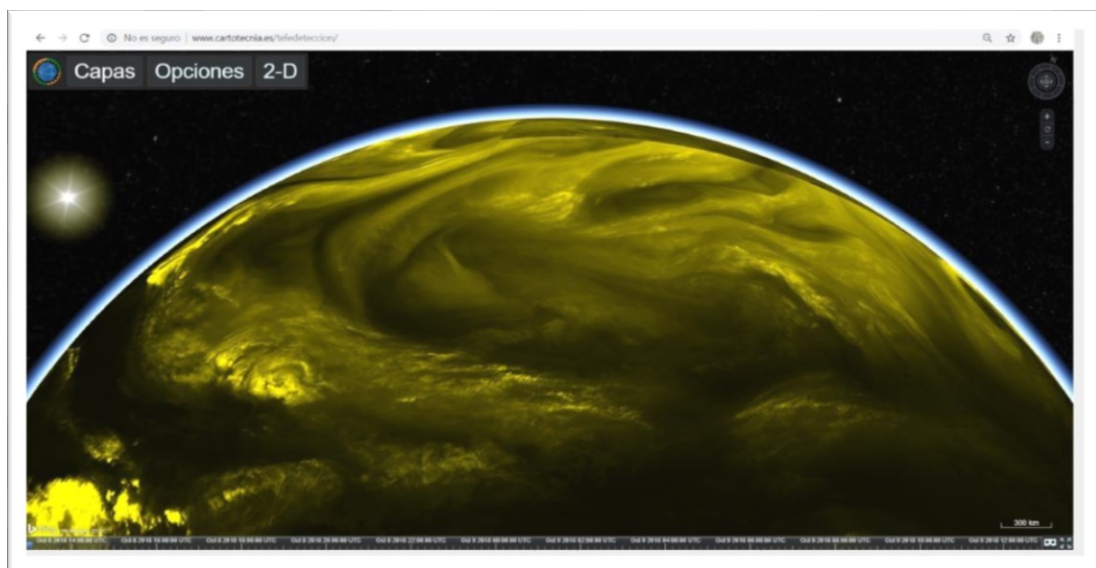


Figura 9.70. Visor online con capa de vapor de agua modificada mediante la herramienta de control de imagen y con la posición relativa del sol a tiempo real
Fuente: elaboración propia a partir de datos de la NASA

Este GIS online permite crear modelos tridimensionales y manipular diferentes capas de información. A este visor se le han incluido las siguientes herramientas:

- Renderizado 2D/3D.
- Renderizado de capas sobre modelo tridimensional del terreno.
- Controles de navegación por pantalla
- Escala
- Orientar al norte
- Navegación con ratón
- Simulación de atmósfera
- Simulación de niebla
- Simulación de superficie de agua.
- Iluminación mediante posición del sol a tiempo real.
- Selector de capas
- Controles de imagen (modificación de las capas ráster a tiempo real para teledetección).

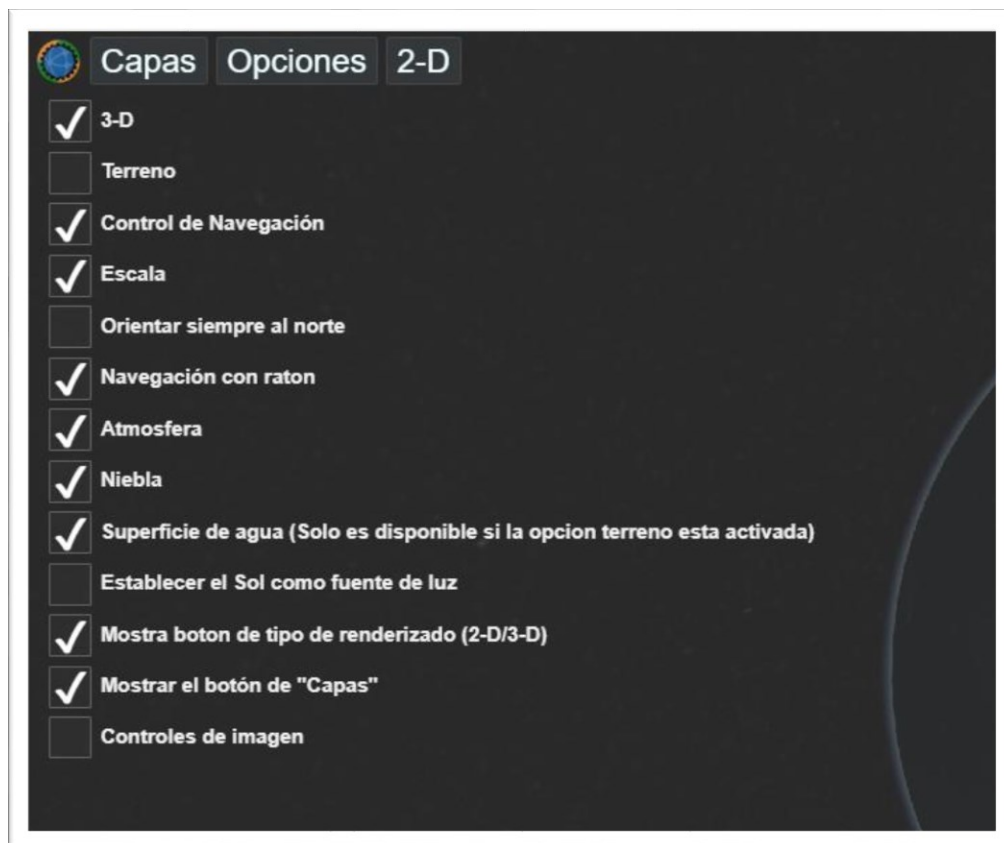


Figura 9.71. Opciones y herramientas del Visor Cartográfico para teledetección con Realidad Aumentada

Fuente: elaboración propia

Los controles de imagen que permiten modificar las capas ráster en tiempo real, actúan sobre los siguientes parámetros de la imagen que compone la capa ráster:

- Transparencia
- Brillo
- Contraste
- Tono
- Saturación
- Gamma

Todas las opciones disponibles en el visor son aplicables tanto a las capas seleccionadas como al mapa base seleccionado.

Para facilitar el uso del software en la labor de estudio del relieve de España, al iniciarse se ubica automáticamente en la península ibérica; sin embargo, a pesar de que la mayor parte de las capas incluidas abarcan el territorio español, este software permite también el estudio del relieve mundial, del cual se han incluido varias capas, entre las que se encuentran imágenes satelitales, aéreas, MDT y hillshades mundiales.



Figura 9.72. Vista inicial del software online desarrollado para el estudio del relieve
Fuente: elaboración propia a partir de datos de Microsoft

El software online, incluye numerosas capas, tanto del territorio de España como mundial, las cuales pueden superponerse y editarse a tiempo real; para ello, en primer lugar, hay que seleccionar las capas que se van a utilizar, mediante el menú de “Capas”:



Figura 9.73. Menú de Capas del software online con realidad virtual
Fuente: elaboración propia

Desde este menú pueden seleccionarse las capas superpuestas y su opacidad; también, mediante las flechas situadas a la derecha, se puede cambiar el orden de visualización de las distintas capas. Para añadir más opciones a las distintas capas y poder modificar algunos de los parámetros de las imágenes que las componen, pueden ampliarse los parámetros modificables, activando la opción “Controles de Imagen”, situada al final del menú “Opciones”. Una vez activada esta opción, pueden modificarse el brillo, el contraste, el tono, la saturación y el gamma, en tiempo real, de una o varias capas mientras se exploran.



Figura 9.74. Imagen de la opción del menú del software online para activar los controles de imagen

Fuente: elaboración propia

Una vez activada esta opción, al acceder al menú de “Capas”, cada una de ellas mostrará los parámetros de imagen mencionados anteriormente.

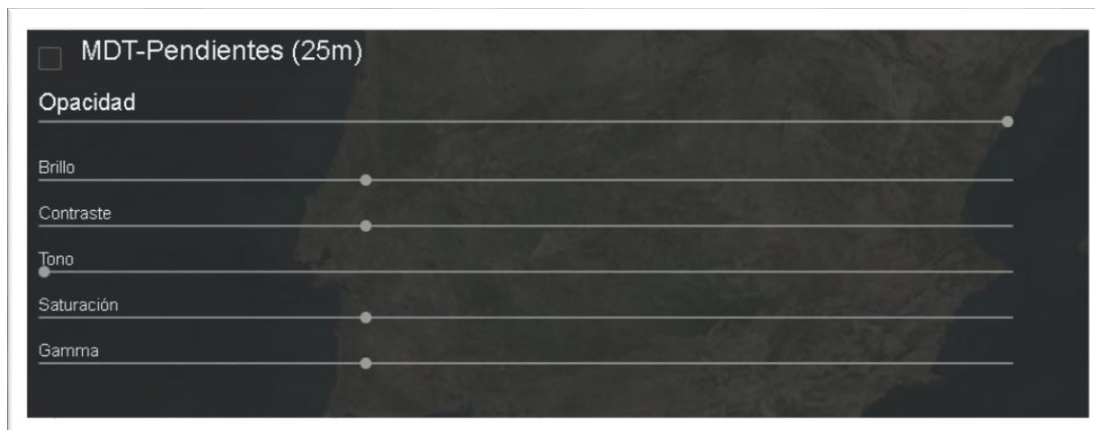


Figura 9.75. Controles de imagen disponibles en el Visor online para Teledetección a tiempo real con Realidad Virtual

Fuente: elaboración propia

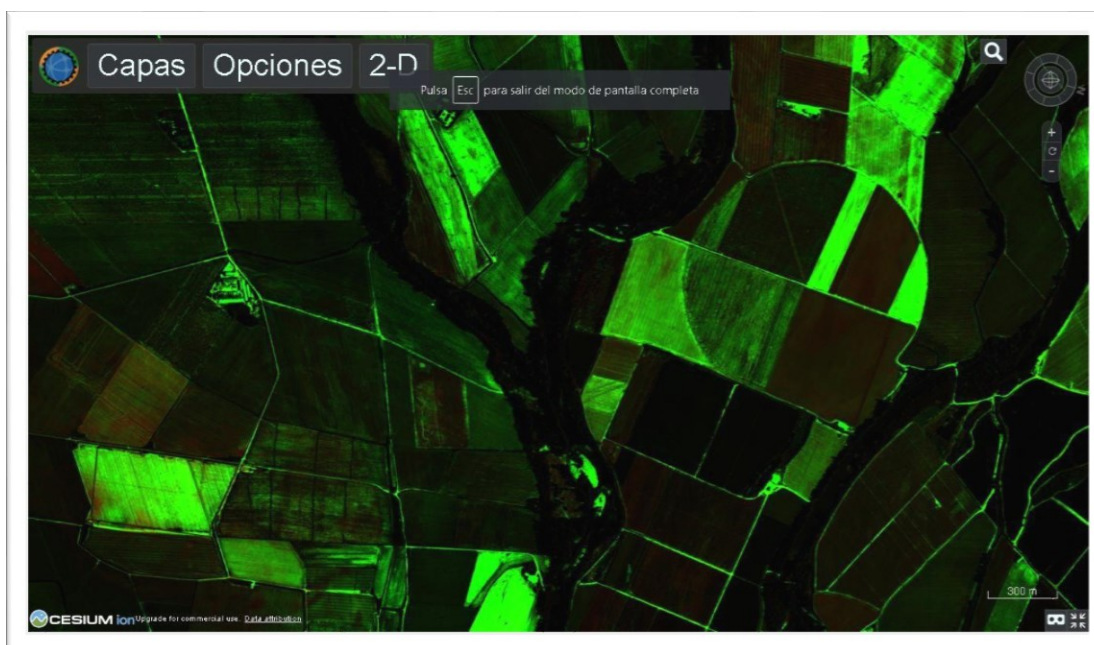


Figura 9.76. Imagen generada con el visor online (a pantalla completa) en la que se han aplicado los filtros incluidos en la opción “controles de imagen” del visor, para la resaltar determinados cultivos y facilitar su teledetección
Fuente: elaboración propia, a partir de datos del IGN

Para maximizar el área de trabajo, puede extenderse el visor mediante el botón situado en la esquina inferior derecha, junto al cual se sitúa la opción activable de Realidad Virtual y las coordenadas WGS 84 en grados decimales.

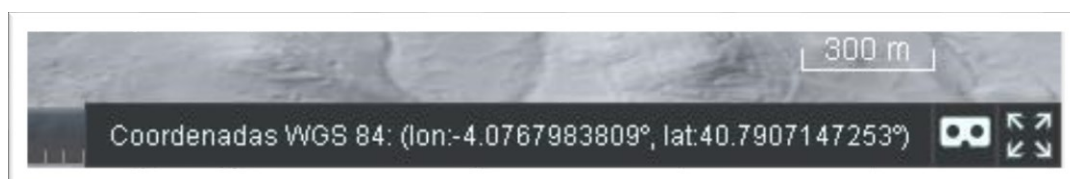


Figura 9.77. Detalle de las coordenadas, escala y botones para la Realidad Virtual o la expansión del espacio de trabajo, en el software online
Fuente: elaboración propia

La escala se adapta en función al zoom aplicado y se sitúa sobre los botones mencionados en el párrafo anterior.

Para modificar la perspectiva tridimensional, se utiliza el movimiento del ratón, mientras se presiona el botón central del mismo. Para facilitar el uso en otros dispositivos, esta perspectiva puede variarse también mediante los controles situados en la esquina superior derecha.

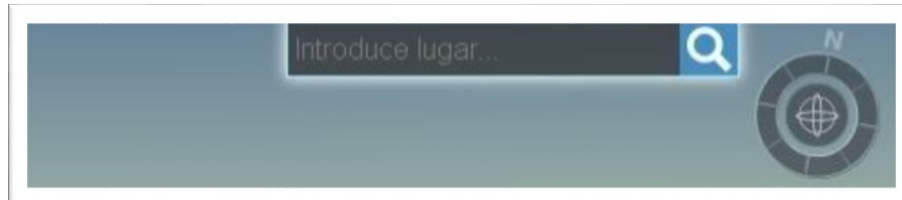


Figura 9.78. Imagen del cuadro de texto para localización de ubicaciones
Fuente: elaboración propia

Estos controles permiten además modificar el zoom y muestran el Norte; a la izquierda de éstos está el botón de búsqueda, mediante el cual se pueden encontrar ubicaciones utilizando la geocodificación, que incluye autocompletado.



Figura 9.79. Imagen de ejemplo del autocompletado del cuadro de texto para búsqueda de lugares en el software online
Fuente: elaboración propia

Para poder observar los relieves tridimensionales con distintas iluminaciones, correspondientes a distintos momentos, puede activarse la opción “Establecer sol como fuente de luz”.

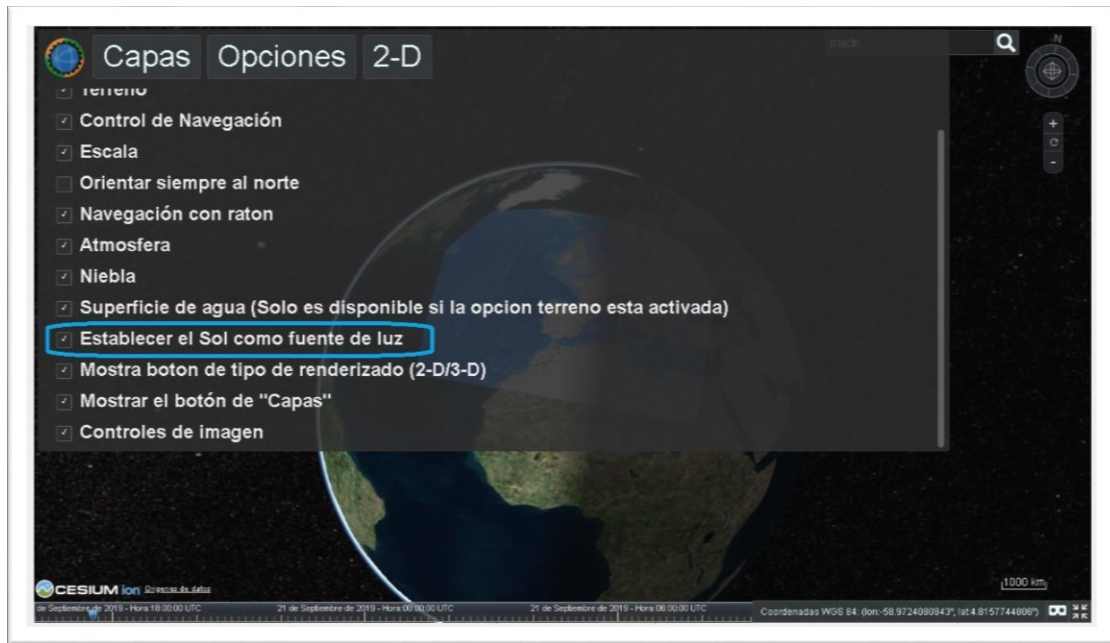


Figura 9.80. Opción de iluminación mediante el sol
Fuente: elaboración propia

Activando esta opción, muestra la iluminación en el momento actual; sin embargo, puede modificarse la fecha o la hora mediante la barra de tiempo situada en el margen inferior del visor.



Figura 9.81. Detalle de la barra de tiempo para configurar la iluminación
Fuente: elaboración propia

Por otra parte, el botón 2D permite modificar la vista para representar los datos en dos dimensiones.



Figura 9.82. SIG de realidad virtual configurado para representación bidimensional
Fuente: elaboración propia, a partir de datos del IGN y de Microsoft

Para acelerar el procesamiento en equipos más antiguos, se puede desactivar la opción de “Terreno”, cargando en su lugar el elipsoide.

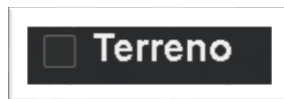


Figura 9.83. Detalle de la opción terreno, accesible desde el menú de opciones
Fuente: elaboración propia

Esta opción de Terreno debe estar activada para mostrar los efectos dinámicos del agua.



Figura 9.84. Ejemplo de la visualización de los efectos dinámicos del agua en el SIG de realidad virtual

Fuente: elaboración propia, a partir de datos del IGN, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia

Una de las utilidades más reseñables que se han incluido en el software online, es la de consultar las propiedades de los elementos contenidos en las distintas capas y así facilitar su interpretación de forma rápida.



Figura 9.85. Información de Simbología del Cuaternario, incluida en la capa de mapa geológico 1:50.000, superpuesta al relieve tridimensional y a la imagen aérea en el SIG de realidad virtual

Fuente: elaboración propia a partir de datos del IGN, el IGME, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia

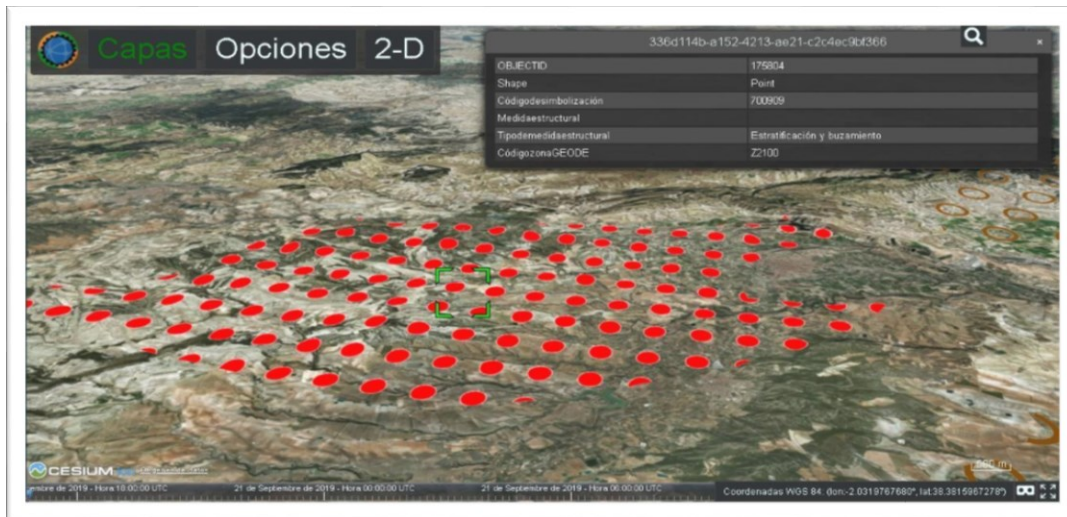


Figura 9.86. Visualización de estratificación y buzamiento (trama de puntos rojos), mediante el SIG de realidad virtual

Fuente: elaboración propia, a partir de datos del IGN, el IGME, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia

En las imágenes mostradas anteriormente pueden observarse algunos ejemplos de utilización del SIG online con Realidad Virtual para la extracción de información, consultando las bases de datos del IGME.

10. Casos de uso del visor VR y MEL

Este capítulo recoge ejemplos de posibles aplicaciones en el ámbito del estudio geomorfológico y de la arqueología, tanto del Visor de Realidad Virtual online, como de la herramienta de escritorio MEL, ambas desarrolladas durante esta investigación.

Mediante las dos herramientas mencionadas anteriormente se han generado las imágenes que se muestran en los distintos subepígrafes de este capítulo. Para poder mostrar las posibilidades que estas dos herramientas ofrecen al análisis geomorfológico se han escogido localizaciones con un especial interés geomorfológico dentro del territorio español. Las diferentes localizaciones utilizadas para generar los ejemplos mostrados en este capítulo han sido extraídas de la capa “Inventario Español de Lugares de Interés Geológico (IELIG)” incluida en el Visor de Realidad virtual online, el cual está disponible a través de la siguiente dirección web “<http://cartotecnia.es/teledeteccion/>”, tal y como se muestra en la siguiente imagen.

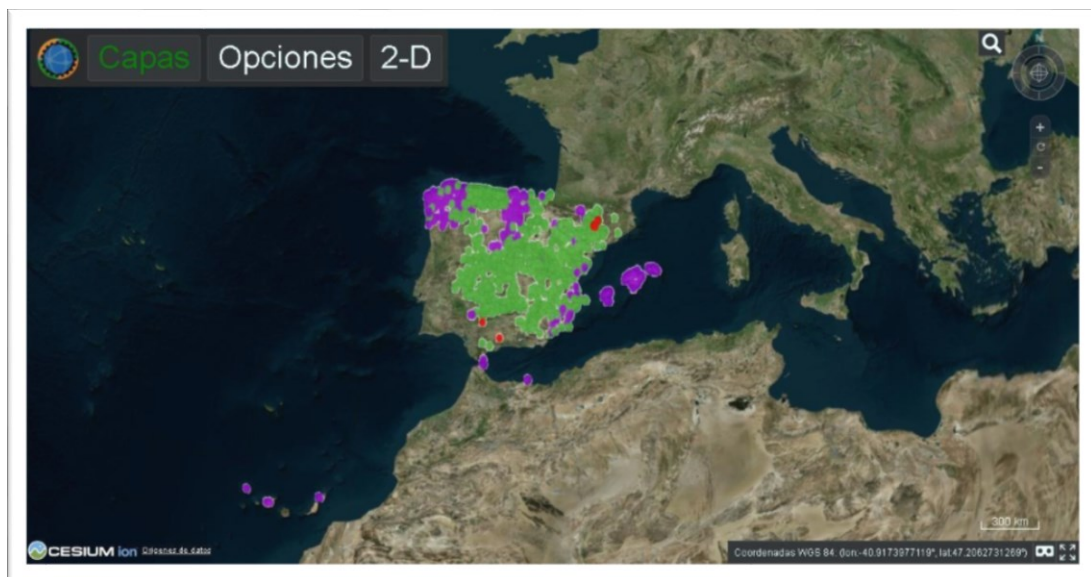


Figura 10.1. Capa de lugares de interés geológico
Fuente: elaboración propia, a partir de datos de Microsoft y del IGME

Gracias a la herramienta de escritorio MEL, pueden eliminarse los puntos correspondientes a vegetación o edificios presentes en las nubes de puntos suministradas por el IGN; de esta forma se pueden generar hillshades, mapas de pendientes y mapas de acumulación de flujo que permiten un estudio detallado del relieve, con una resolución mucho mayor a la proporcionada por los MDT del IGN.

Mediante la herramienta online que se ha desarrollado, además, puede accederse a los datos asociados a distintas localizaciones en aquellas capas que dispongan de dichos datos. Un ejemplo de consulta a los datos asociados a una de las localizaciones contenidas en la capa “Inventario Español de Lugares de Interés Geológico (IELIG)” puede verse en la siguiente imagen.



Figura 10.2. Ejemplo de consulta de datos de uno de los lugares incluidos en la capa de lugares de interés geológico

Fuente: elaboración propia a partir de datos del IGN y del IGME

A continuación, se muestra una ampliación de algunos de los datos a los que es posible acceder mediante la consulta al IELIG, a través del Visor de Realidad Virtual.

Denominación	La Tarta
CodConfidencialidad	1
Descripcion	Forma parte de las últimas erupciones piroclásticas de la Dorsal, con el nombre de "tarta", formada por una alternancia de piroclastos básicos y sálcos emitidos en los últimos episodios eruptivos.
MostrarWeb	1
InteresPrincipal	Petrológico-geoquímico
InteresSecundario	Null
UnidadGeologica	Sistemas volcánicos recientes
ContextoGeologico	Edificios y morfologías volcánicas de las Islas Canarias

Figura 10.3. Detalle de parte de la información de los lugares de interés geológico, accesible desde el software online desarrollado en este trabajo
Fuente: elaboración propia, a partir de datos del IGME

Para realizar un estudio más detallado, el SIG de Realidad Virtual desarrollado permite consultar los detalles de cualquiera de los distintos elementos, accediendo a las bases de datos del IGME, tal y como se muestra en el siguiente ejemplo, en el cual se realiza una consulta de un elemento presente en la capa “Recintos geología-Mapa Geológico Continuo de España a escala 1:50.000”; concretamente, se trata de depósitos sedimentarios indeterminados del Holoceno, situados en la caldera de Las Cañadas, en Tenerife.

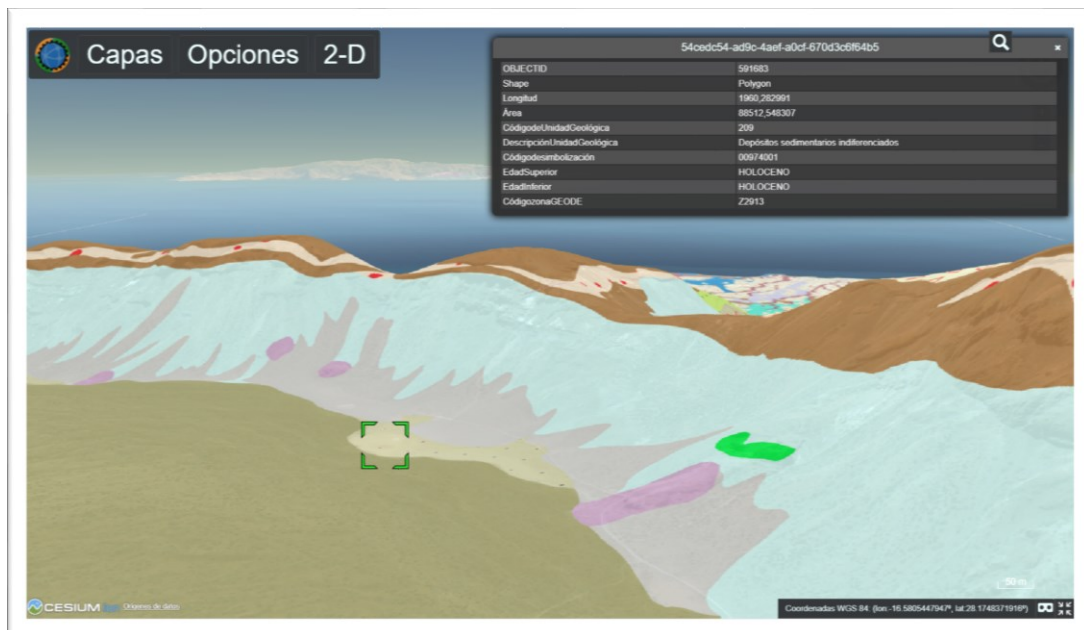


Figura 10.4. Consulta de elemento geológico mediante el SIG online de Realidad Virtual
Fuente: elaboración propia, a partir de datos del IGME, Microsoft, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia

10.1. Valle de Ucanca (Santa Cruz de Tenerife)

Nombre del archivo LiDAR utilizado	PNOA_2016_CANAR-TF_338-3124_ORT-CLA-RGB.laz
X (UTM ETRS89)	339756
Y (UTM ETRS89)	3123093
Huso	28

Este ejemplo muestra parte del Llano de Ucanca, situado en el municipio de la Orotava, donde pueden apreciarse llamativas estructuras como los roques monolíticos y las coladas de lava procedentes del Teide. Si bien estas estructuras pueden llegar a ser distinguibles mediante el uso de ortofotografías, los modelos tridimensionales de nube de puntos, los hillshades, los mapas de pendientes y de acumulación de flujo, generados mediante la aplicación de escritorio MEL aquí desarrollada, las muestran en todo su esplendor gracias al modelado tridimensional y la resolución de 25 centímetros por píxel.

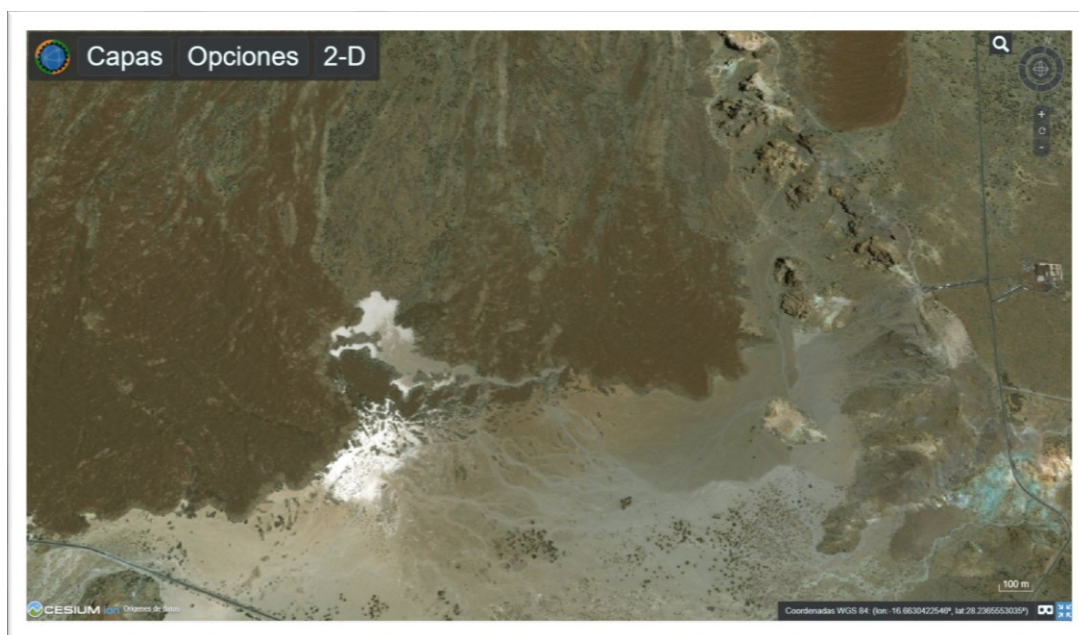


Figura 10.5. Valle de Ucanca, montaña de Roque y los Azulejos (derecha)
Fuente: elaboración propia, a partir de datos de Microsoft

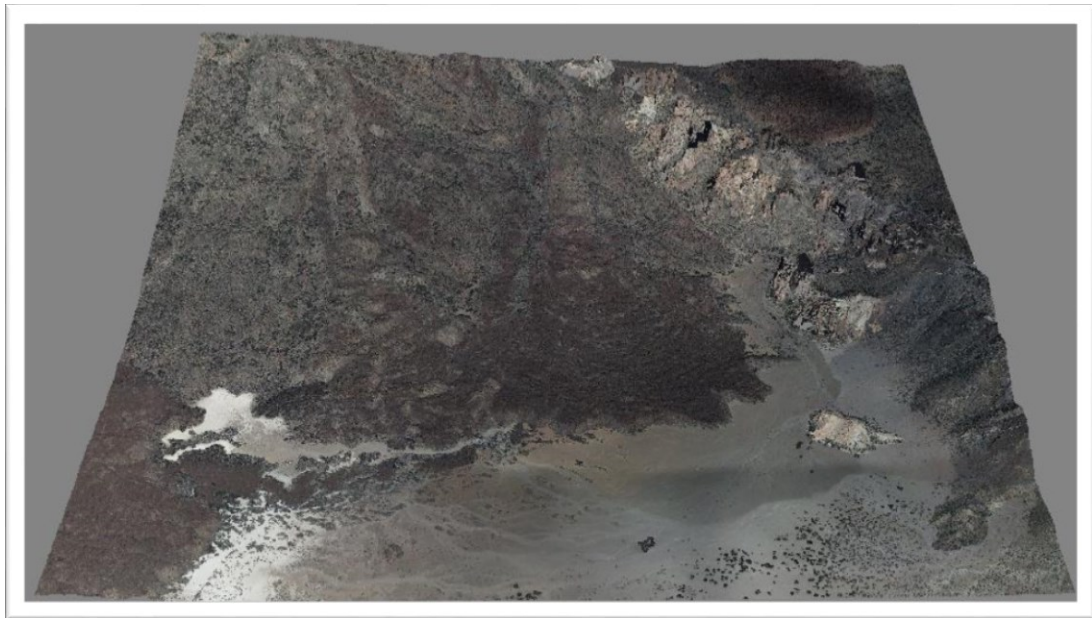


Figura 10.6. Modelo tridimensional de nube de puntos del archivo PNOA_2016_CANAR-TF_338-3124_ORT-CLA-RGB.laz del IGN, generado con el software MEL, con color RGB
Fuente: elaboración propia, a partir de datos del IGN

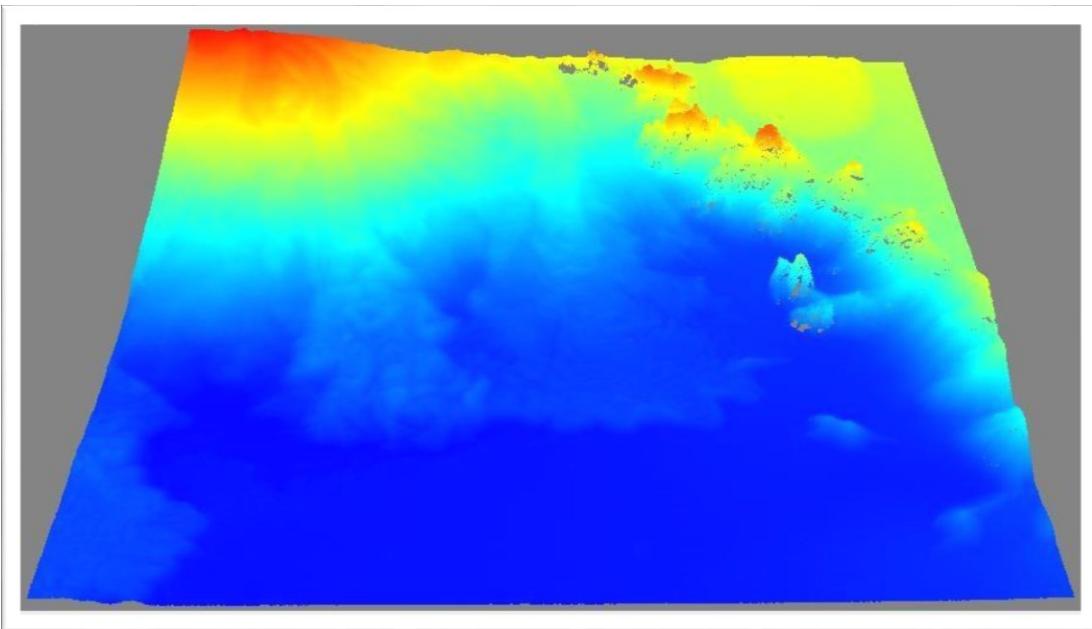


Figura 10.7. Modelo tridimensional de la nube de puntos del archivo PNOA_2016_CANAR-TF_338-3124_ORT-CLA-RGB.laz del IGN, generado con el software MEL, con coloración por alturas
Fuente: elaboración propia, a partir de datos del IGN

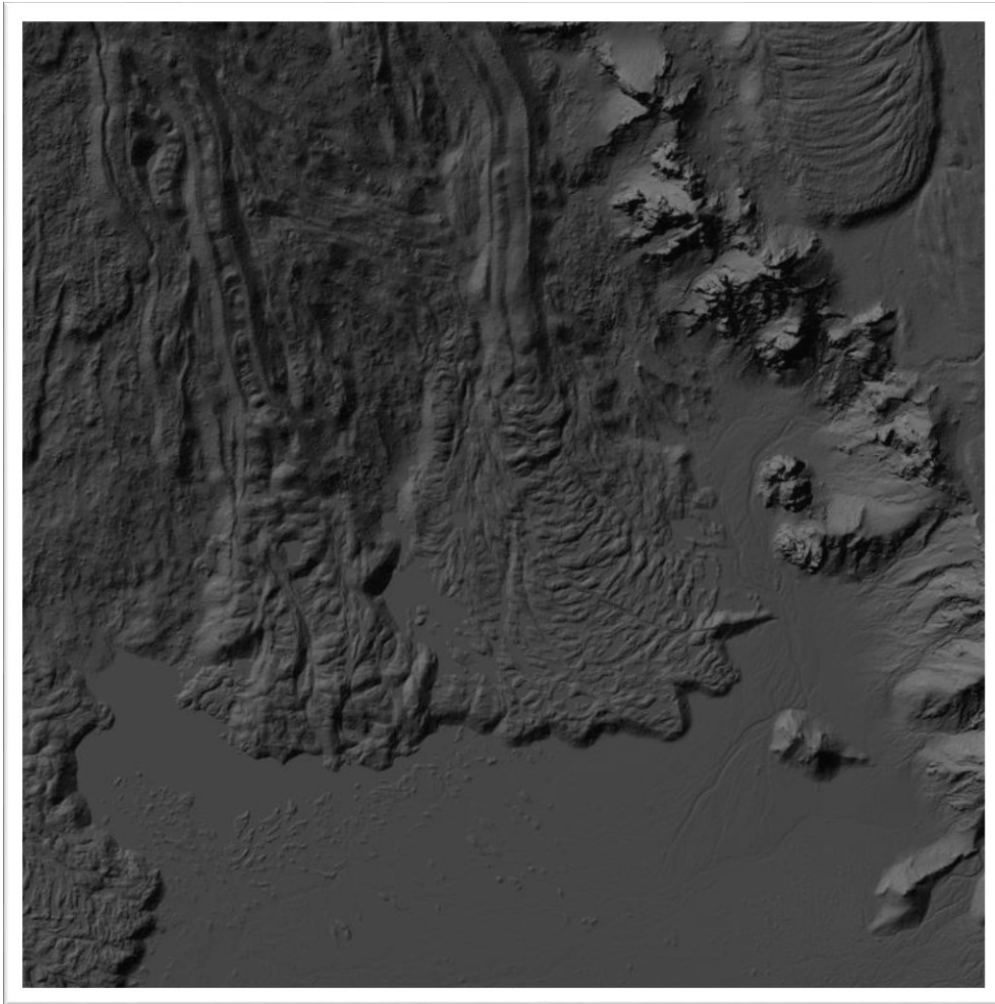


Figura 10.8. Hillshade de las coladas del Teide con resolución de malla de 25 cm, generado con el software MEL.

Fuente: elaboración propia, a partir de datos del IGN

En la zona superior del hillshade (Figura 10.8), pueden apreciarse tanto coladas cordadas (derecha), como tubos de lava (centro e izquierda). También son apreciables depósitos por deslizamiento (centro).

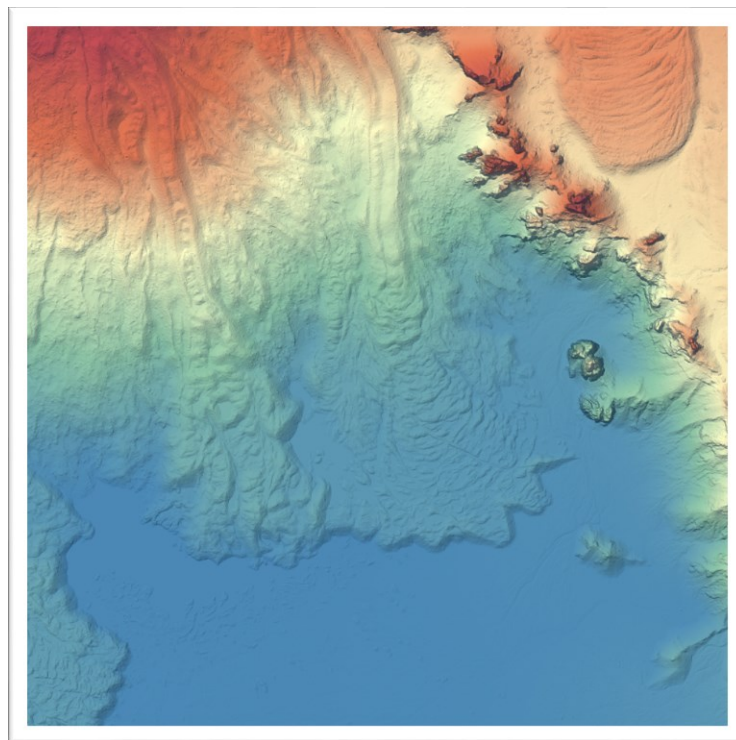


Figura 10.9. Mapa hipsométrico de las coladas del Teide con resolución de malla de 25 cm, generado con el software MEL (paleta: muted).
Fuente: elaboración propia, a partir de datos del IGN

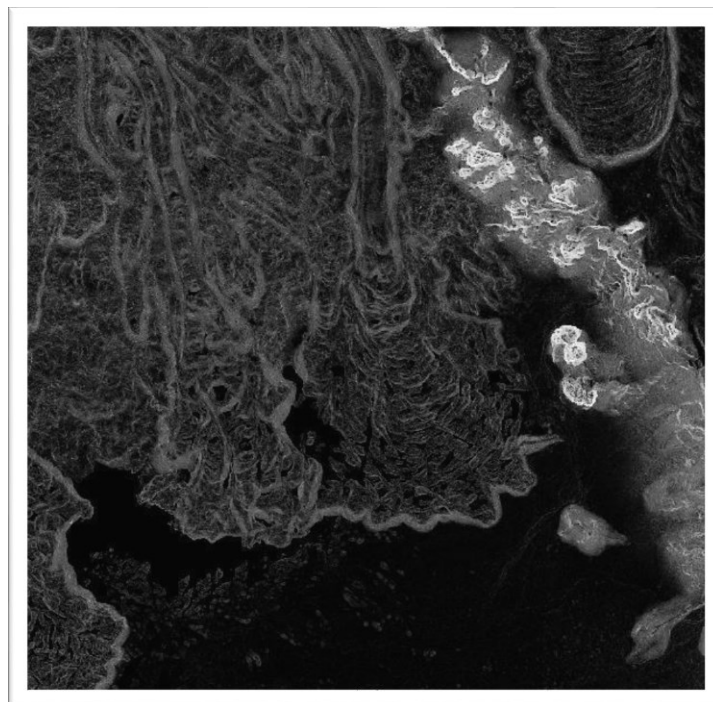


Figura 10.10. Mapa de pendientes de las coladas del Teide con resolución de malla de 25 cm, generado con el software MEL
Fuente: elaboración propia, a partir de datos del IGN

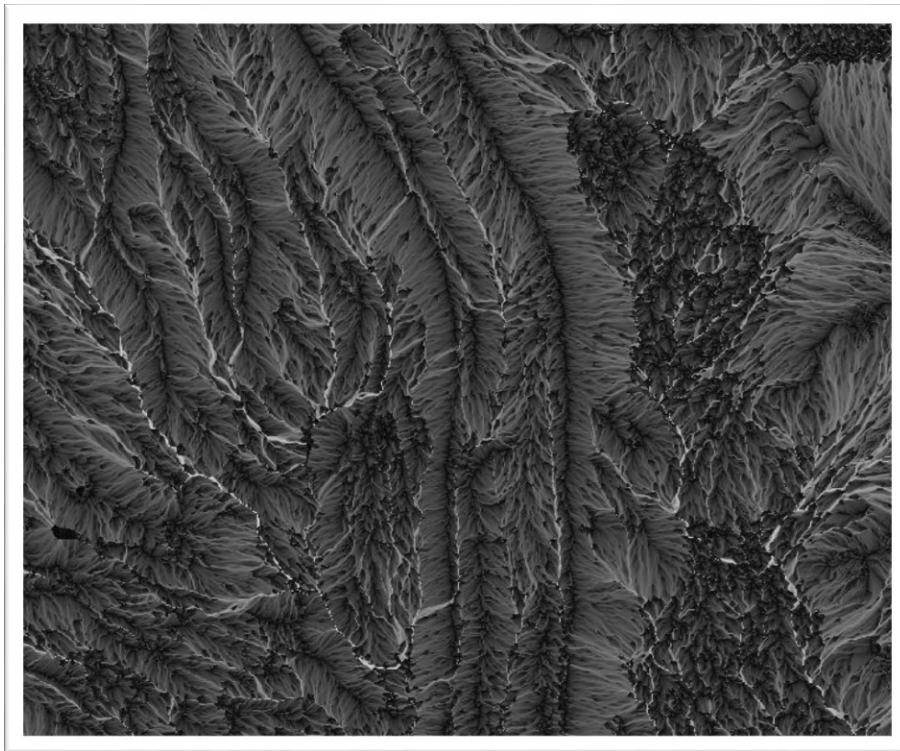


Figura 10.11. Mapa de acumulación de flujo de las coladas del Teide con resolución de malla de 25 cm, generado con el software MEL

Fuente: elaboración propia, a partir de datos del IGN

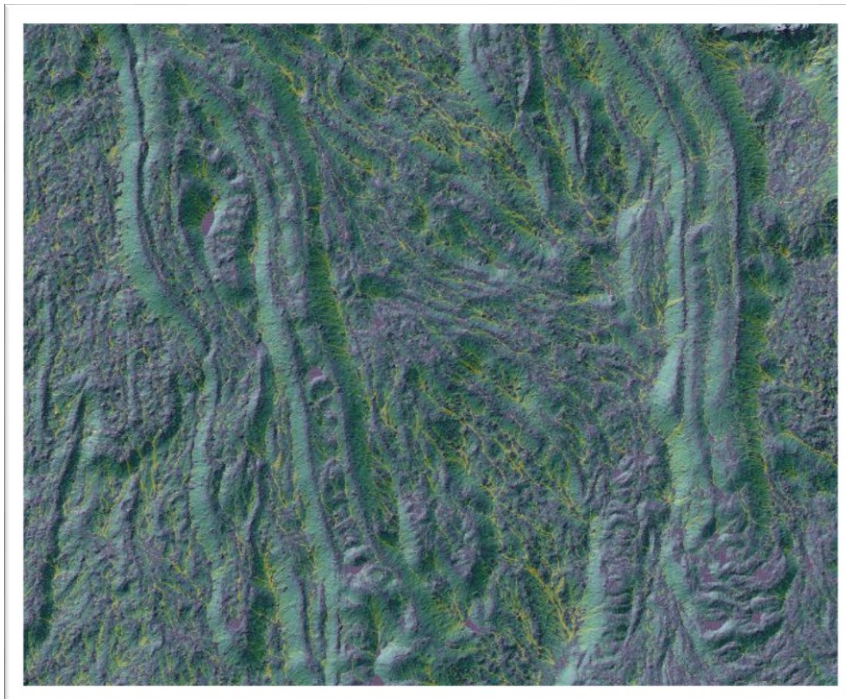


Figura 10.12. Detalle del hillshade con coloración derivada del mapa de acumulación de flujo con pseudocolor monobanda de las coladas del Teide, con resolución de malla de 25 cm; las distintas capas han sido generadas con el software MEL y combinadas con QGIS

Fuente: elaboración propia, a partir de datos del IGN

10.2. Presa y cascadas de Ézaro (La Coruña)

Nombre del archivo LiDAR utilizado	PNOA-2010-LOTE10-GAL-490-4752-ORT-CLA-COL.LAZ
X (UTM ETRS89)	490877
Y (UTM ETRS89)	4751394
Huso	29

Este ejemplo recoge algunas de las geomorfologías presentes en los últimos kilómetros del río Xallas. En las imágenes mostradas a continuación puede apreciarse cómo el software desarrollado es capaz de representar los datos tridimensionales con suficiente precisión como para que sean claramente apreciables tanto las morfologías naturales (como las cascadas) y antropogénicas (como la presa del Embalse de Ézaro). Tanto en el hillshade, el mapa hipsométrico, los mapas de pendiente o las representaciones tridimensionales de nubes de puntos mostrados a continuación pueden apreciarse otras formaciones como gargantas, cañones y hoces.



Figura 10.13. Vista del modelo tridimensional del curso bajo del Xallas, con textura de Microsoft Satélite, generada con el software online para el estudio del relieve
Fuente: elaboración propia, a partir de datos del IGN, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia

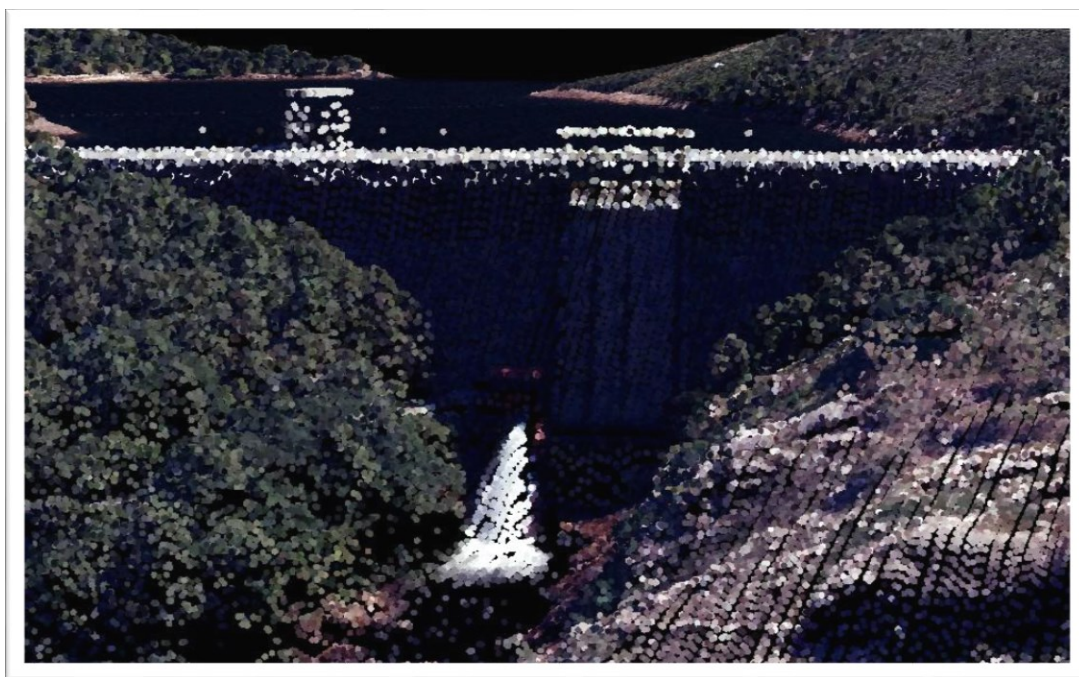


Figura 10.14. Detalle de la presa en modelo tridimensional de nube de puntos, generado con el software MEL
Fuente: elaboración propia, a partir de datos del IGN

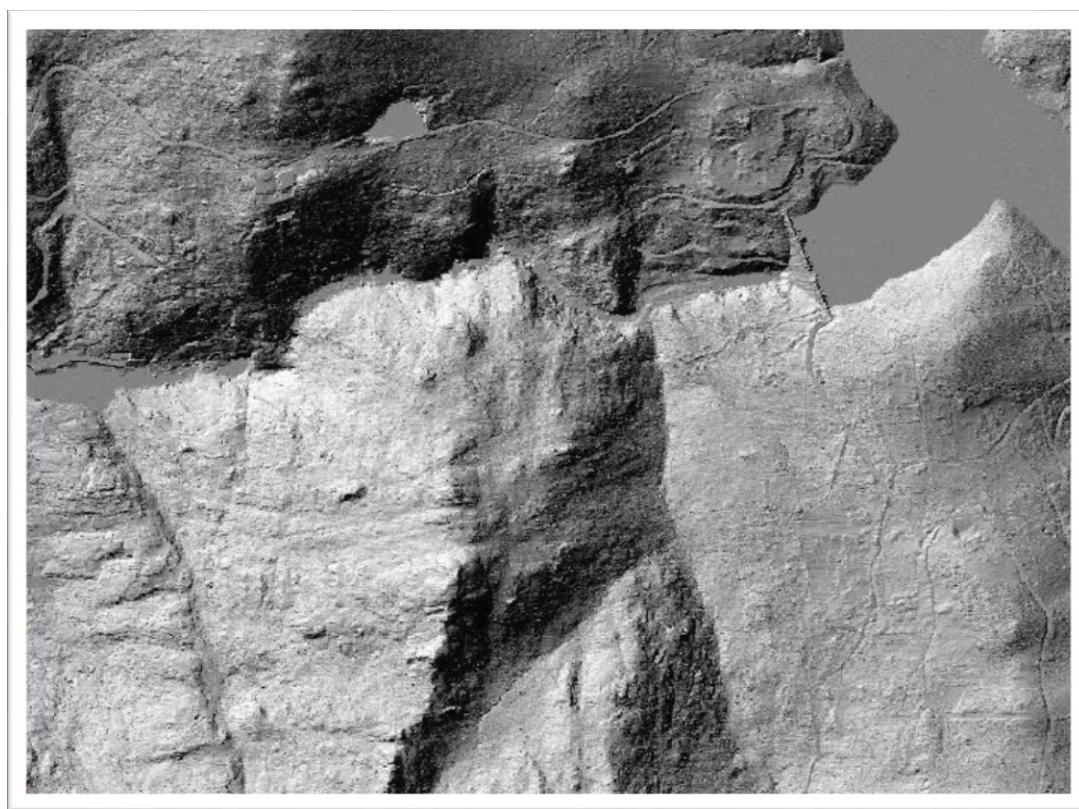


Figura 10.15 Detalle de la presa en hillshade de 25 cm de resolución de malla, generado con el software MEL
Fuente: elaboración propia, a partir de datos del IGN

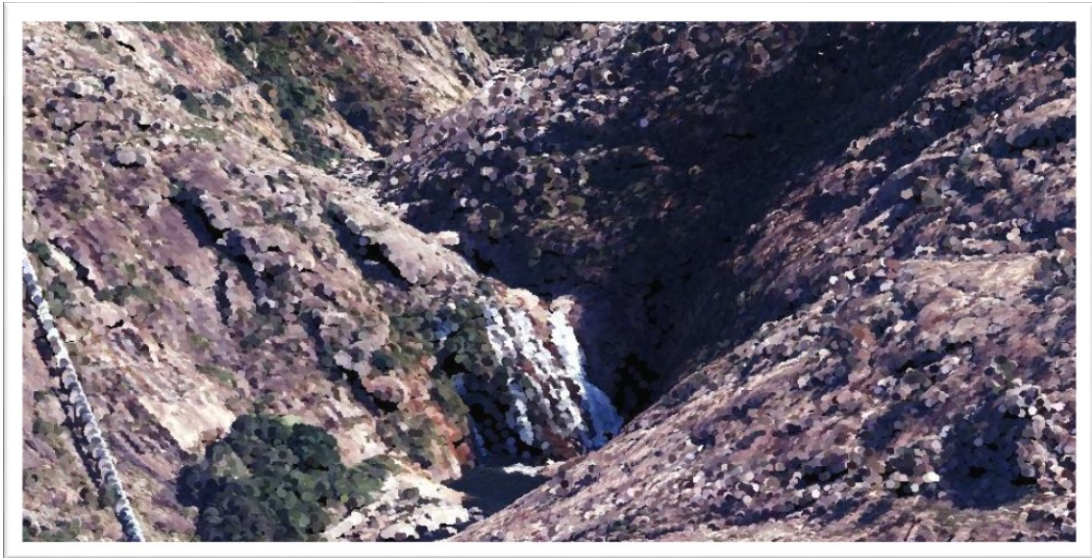


Figura 10.16. Detalle de cascadas en modelo tridimensional de nube de puntos, generado con el software online para el estudio del relieve
Fuente: elaboración propia, a partir de datos del IGN

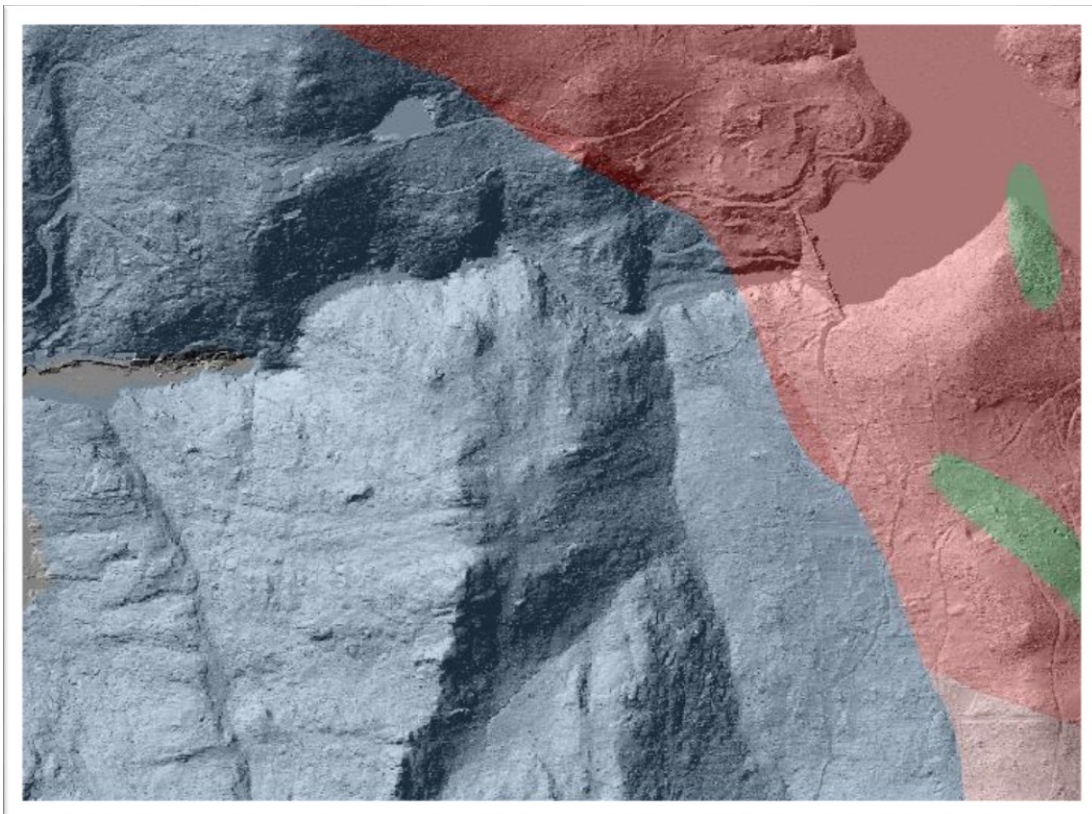


Figura 10.17. Hillshade del curso bajo del Xallas generado con el software MEL coloreado con litologías del Mapa Geológico Nacional del IGME, MAGNA 50
Fuente: elaboración propia, a partir de datos del IGN y del IGME



Figura 10.18. Contexto del hillshade generado con el software MEL, superpuesto a ortomágenes del PNOA

Fuente: elaboración propia, a partir de datos del IGN

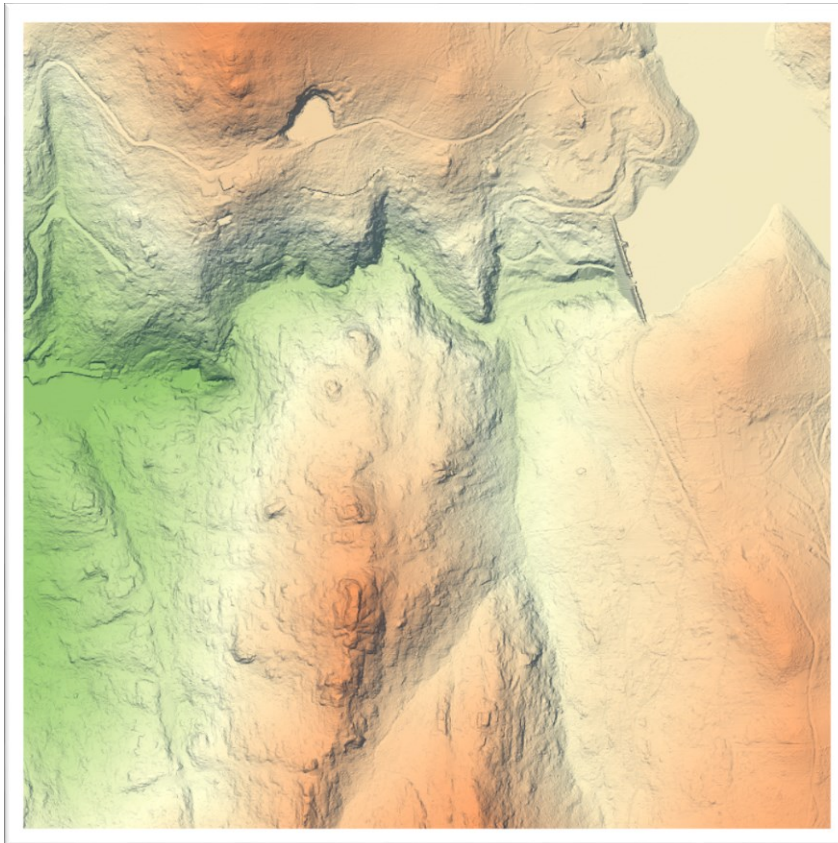


Figura 10.19. Mapa hipsométrico del curso bajo del Xallas con resolución de malla de 25 cm, generado con el software MEL (paleta: atlas).

Fuente: elaboración propia, a partir de datos del IGN

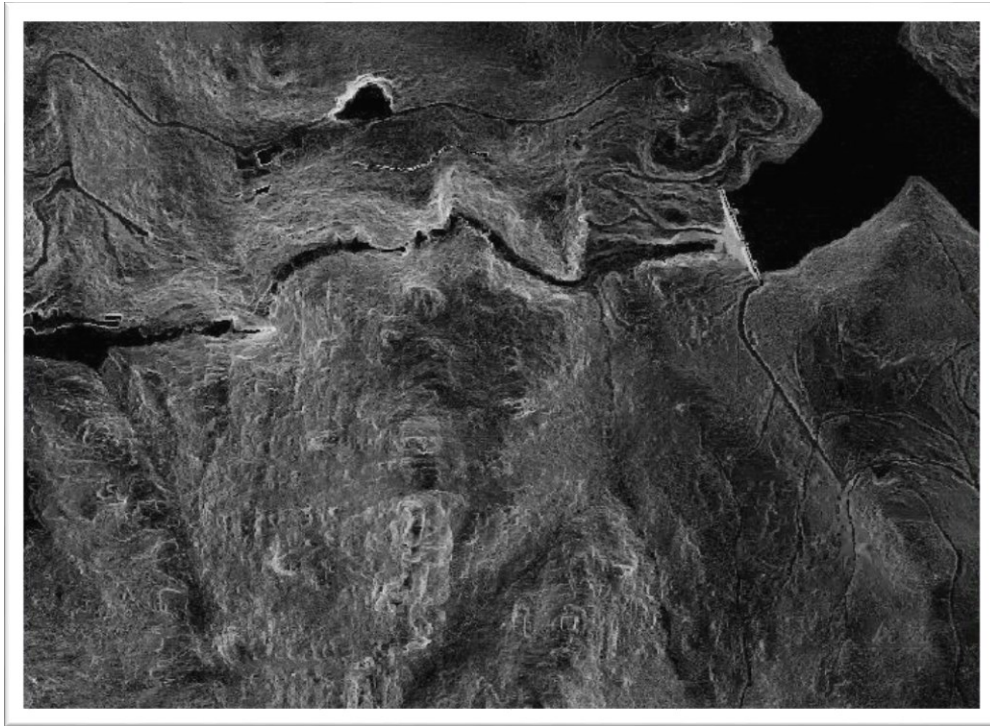


Figura 10.20. Mapa de pendientes del curso bajo del Xallas, con resolución de malla de 25 cm por píxel, generado con el software MEL
Fuente: elaboración propia, a partir de datos del IGN

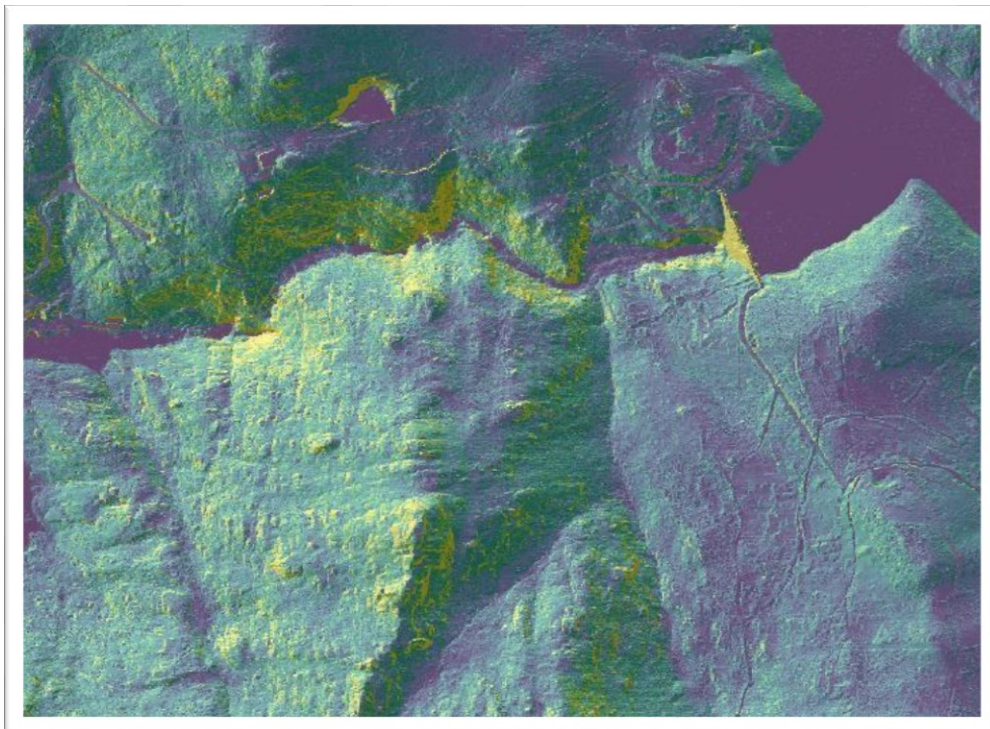


Figura 10.21. Hillshade del curso bajo del Xallas, con resolución de 25 cm por píxel, coloreado con mapa de pendientes en pseudocolor monobanda, creado a partir de la combinación de archivos GeoTiff generados con el software MEL
Fuente: elaboración propia, a partir de datos del IGN

10.3. Raña de Anchuras (Toledo)

Este ejemplo muestra representaciones tridimensionales generadas tanto con el software online de realidad virtual, como con el software de escritorio MEL. Además de los modelos tridimensionales mostrados, se incluye una vista más detallada de las rañas mediante hillshade, mapa de pendientes, mapa hipsométrico y mapa de acumulación de flujo, todos ellos con una resolución de 25 centímetros por pixel, superando ampliamente la máxima resolución ofrecida en los MDT del IGN, los cuales sólo alcanzan una resolución de 2 metros por pixel. A través de las imágenes generadas pueden apreciarse, perfectamente, la red fluvial y las rañas cercanas al municipio de Valdeazores.

Nombre del archivo LiDAR utilizado	PNOA_2009_Lote7_CLM_354-4370_ORT-CLA-COL.laz
X (UTM ETRS89)	870323
Y (UTM ETRS89)	4374281
Huso	29

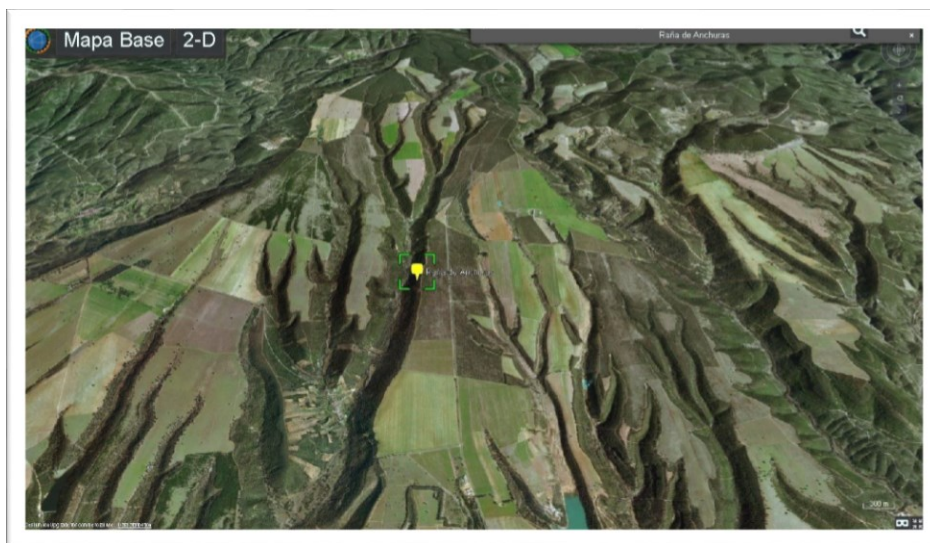


Figura 10.22. Modelo tridimensional de la Raña de Anchuras, generado con el software online para el estudio del relieve, con texturas de Microsoft Satélite

Fuente: elaboración propia, a partir de datos del IGN, el IGME, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia



Figura 10.23. Modelo tridimensional de nube de puntos, con color RGB del archivo PNOA_2009_Lote7_CLM_354-4370_ORT-CLA-COL.laz del IGN, generado con el software MEL

Fuente: elaboración propia, a partir de datos del IGN

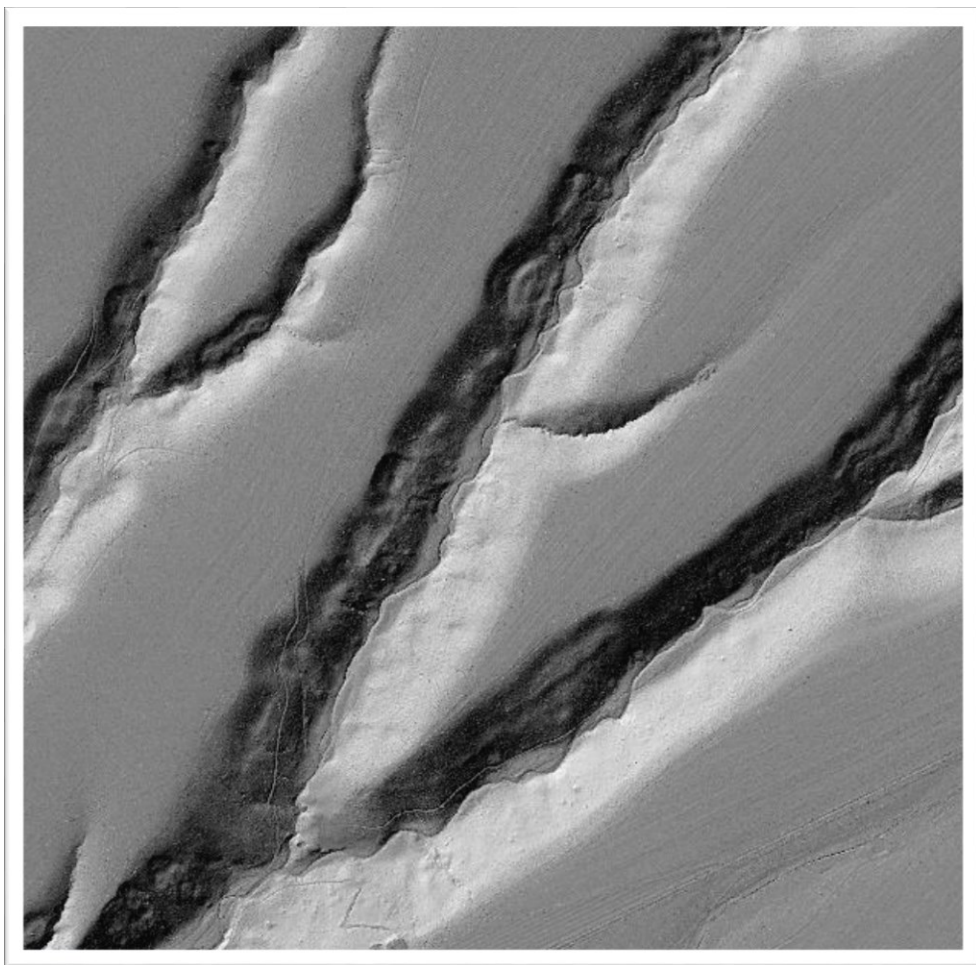


Figura 10.24. Hillshade de la Raña de Anchuras con resolución de malla de 25 cm, generado con el software MEL a partir del archivo PNOA_2009_Lote7_CLM_354-4370_ORT-CLA-COL.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN

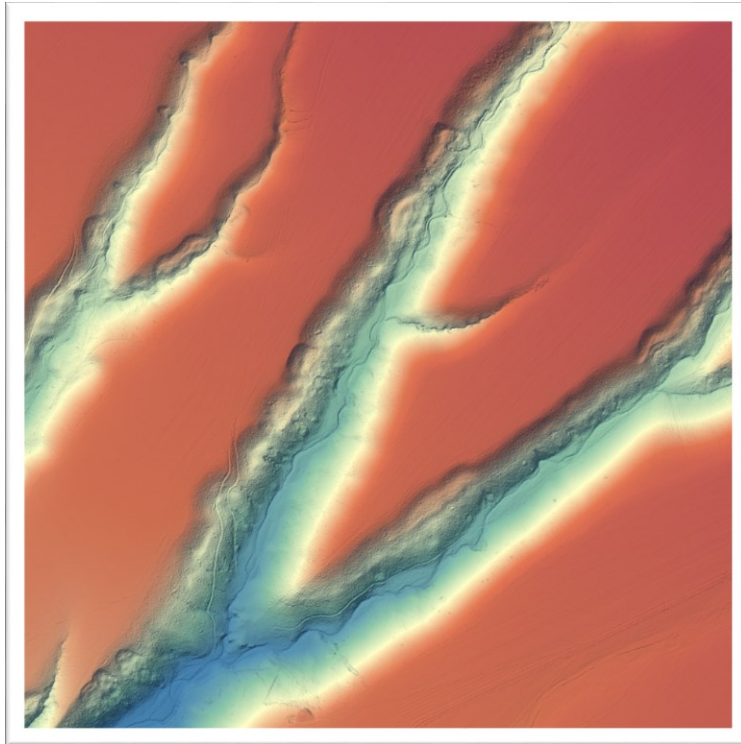


Figura 10.25. Mapa hipsométrico de la Raña de Anchuras con resolución de malla de 25 cm, generado con el software MEL (paleta: muted).
Fuente: elaboración propia, a partir de datos del IGN

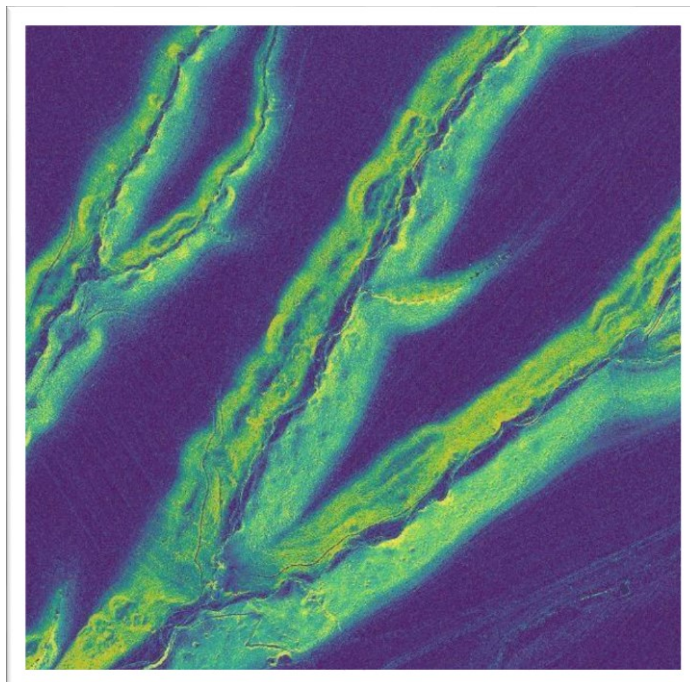


Figura 10.26. Hillshade de la Raña de Anchuras, con resolución de malla de 25 cm con pseudocolor monobanda, generado con el software MEL a partir del archivo PNOA_2009_Lote7_CLM_354-4370_ORT-CLA-COL.laz del IGN
Fuente: elaboración propia, a partir de datos del IGN

10.4. Karst en yesos de Sorbas (Almería)

Nombre del archivo LiDAR utilizado	PNOA-2014-AND-SE-582-4108-ORT-CLA-COL.LAZ
X (UTM ETRS89)	583265
Y (UTM ETRS89)	4106500
Huso	30

En este ejemplo se muestra el karst en yesos de Sorbas, situado en la localidad de Sorbas al Noreste de la provincia de Almería. El karst de yesos de Sorbas constituye uno de los modelos más espectaculares de karst yesíferos del mundo, cuyo desarrollo ha sido posible debido a las rocas evaporíticas (yesos) tan frecuentes en ese territorio. Tanto en los modelos tridimensionales como en los hillshades, en los mapas hipsométricos y en los mapas de pendientes mostrados a continuación, todos ellos generados con herramientas propias, pueden apreciarse distintas estructuras de erosión kárstica, y las llamativas rañas que recorren la extensión mostrada en las siguientes imágenes.

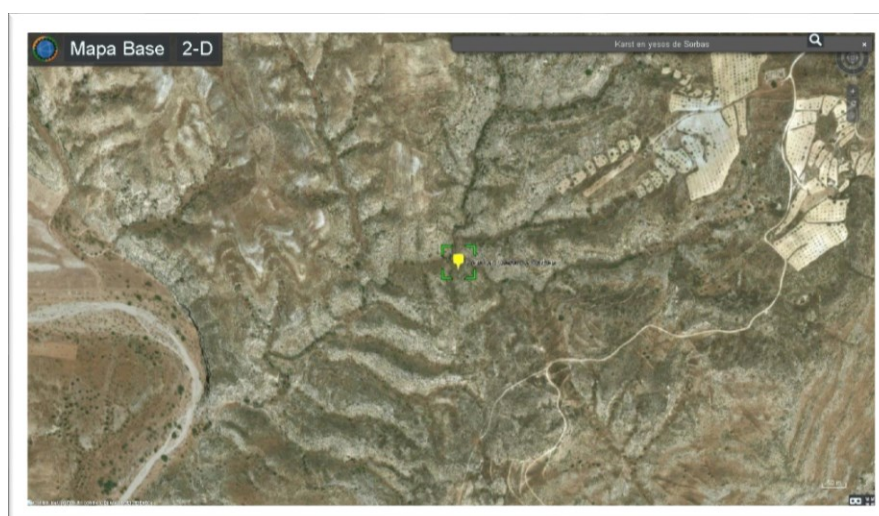


Figura 10.27. Modelo tridimensional del karst en yesos de Sorbas, generado con el software online para el estudio del relieve, con texturas de Microsoft Satélite

Fuente: elaboración propia, a partir de datos del IGN, el IGME, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia



Figura 10.28. Detalle de erosión kárstica en modelo tridimensional de la nube de puntos del karst en yesos de Sorbas, generado con el software MEL con color RGB
Fuente: elaboración propia, a partir de datos del IGN

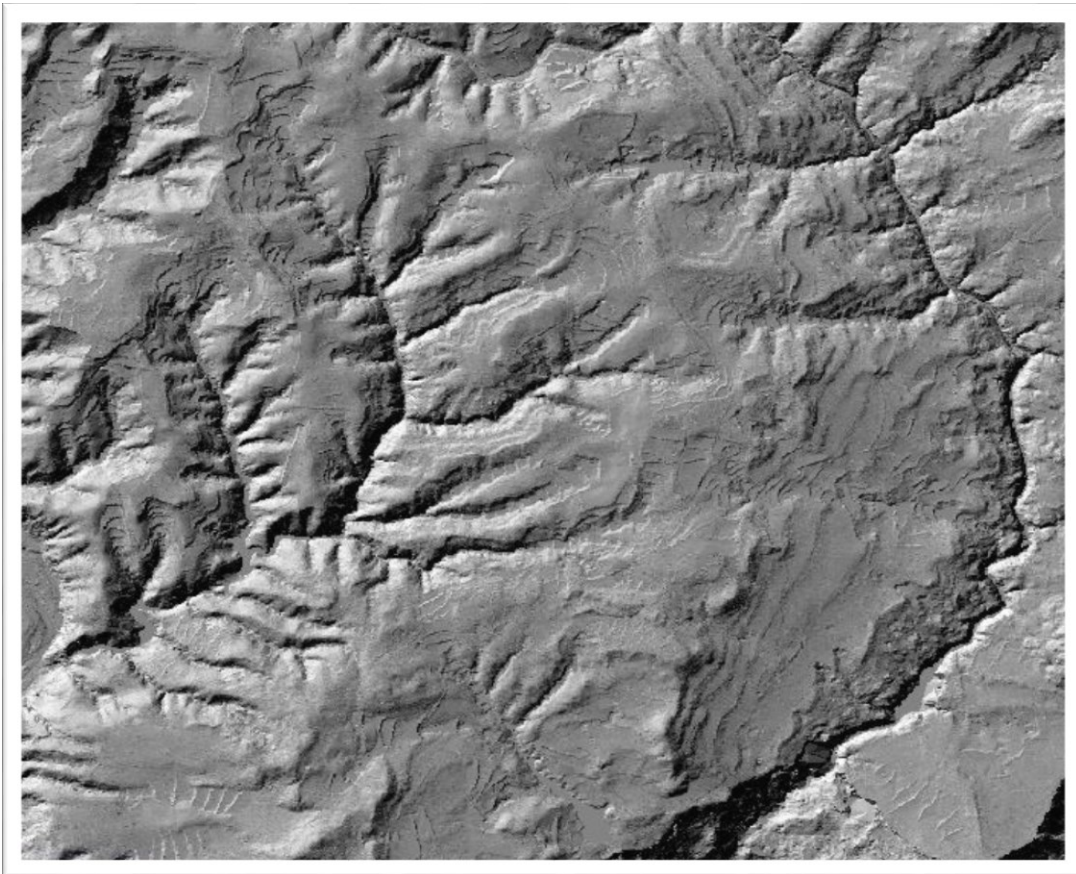


Figura 10.29. Hillshade del karst en yesos de Sorbas con resolución de malla de 25 cm, generado con el software MEL
Fuente: elaboración propia, a partir de datos del IGN

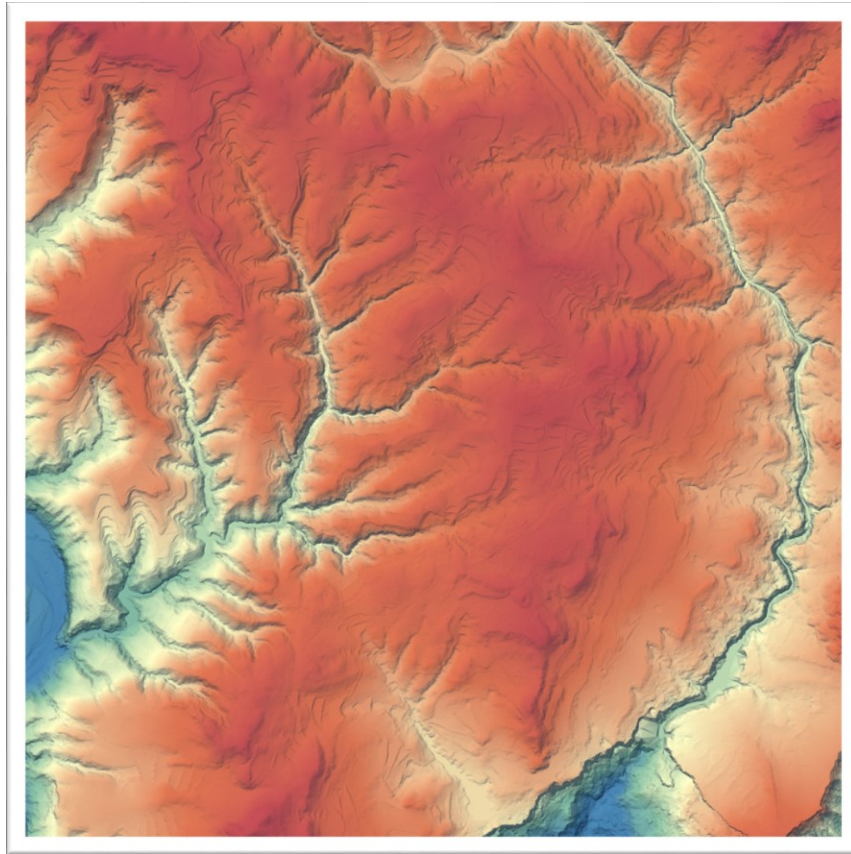


Figura 10.30. Mapa hipsométrico del karst en yesos de Sorbas con resolución de malla de 25 cm, generado con el software MEL (paleta: muted)
Fuente: elaboración propia, a partir de datos del IGN

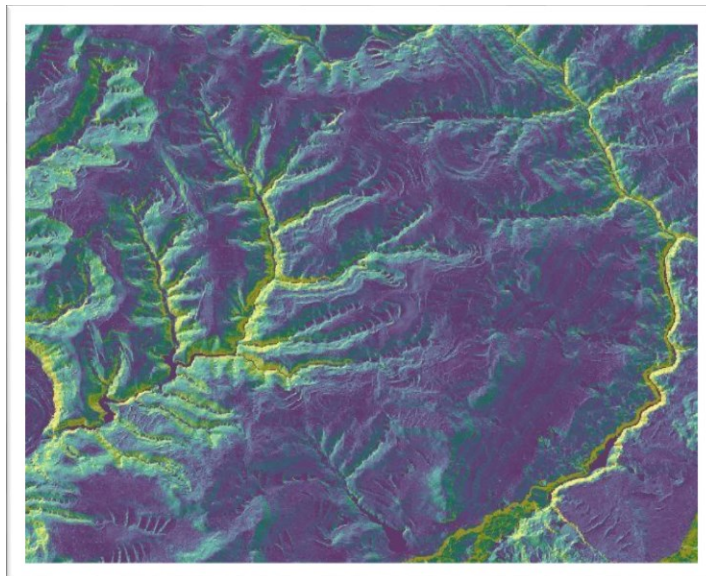


Figura 10.31. Mapa de pendientes del Karst en yesos de Sorbas con resolución de malla de 25 cm y pseudocolor monobanda, generado con el software MEL
Fuente: elaboración propia, a partir de datos del IGN

10.5. Edificio del Teide (Santa Cruz de Tenerife)

Nombre del archivo LiDAR utilizado	PNOA_2016_CANAR-TF_338-3130_ORT-CLA-RGB.laz
X (UTM ETRS89):	339978
Y (UTM ETRS89):	3128824
Huso	28

En este ejemplo se muestra el Edificio del Teide, que tiene la morfología cónica típica de un estratovolcán y se encuentra situado en el municipio de la Orotava. En las imágenes generadas mediante el uso del software creado pueden distinguirse numerosas morfologías propias de los entornos volcánicos, como el cráter semicircular que tiene el Pico en su cima y las numerosas coladas que recorren sus laderas. A simple vista puede observarse cómo mejoran las posibilidades de análisis geomorfológico al utilizar las imágenes obtenidas con el software MEL, en especial con el hillshade, el mapa hipsométrico, el mapa de pendientes y el mapa de acumulación de flujo.

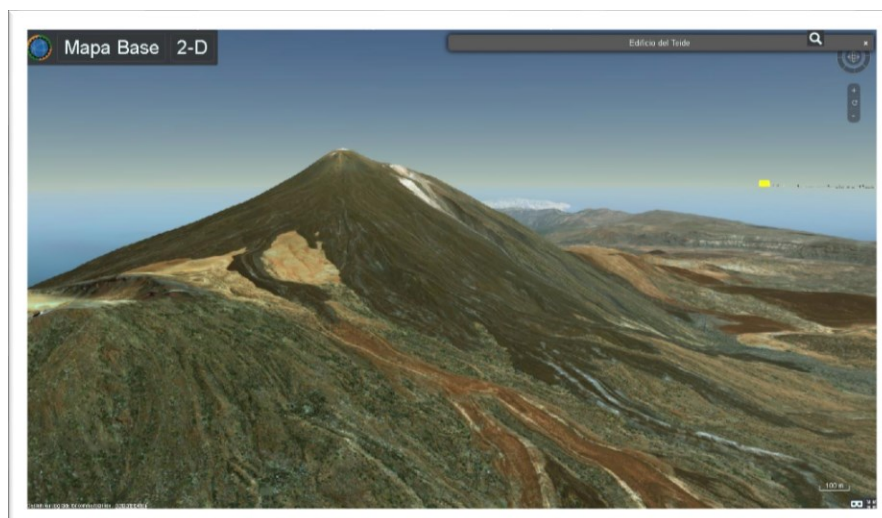


Figura 10.32. Modelo tridimensional del edificio del Teide, generado con el software online para el estudio del relieve, con texturas de Microsoft Satélite

Fuente: elaboración propia, a partir de datos de Microsoft, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia

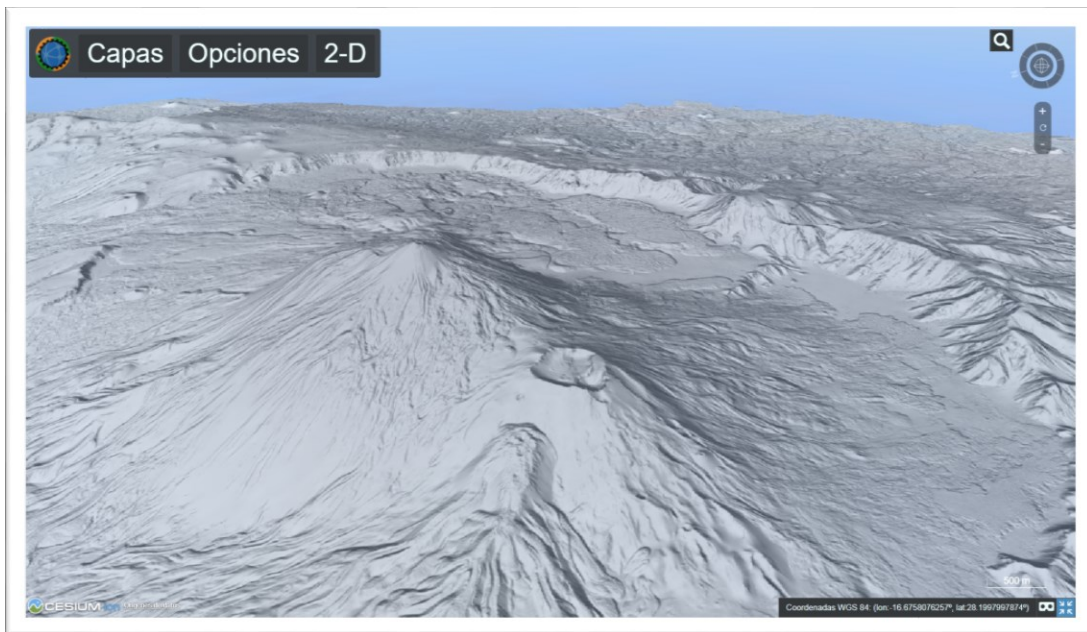


Figura 10.33. Modelo tridimensional del edificio del Teide, generado con el software online para el estudio del relieve, con texturas del mapa del relieve del IGN
Fuente: elaboración propia, a partir de datos del IGN, el IGME, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia

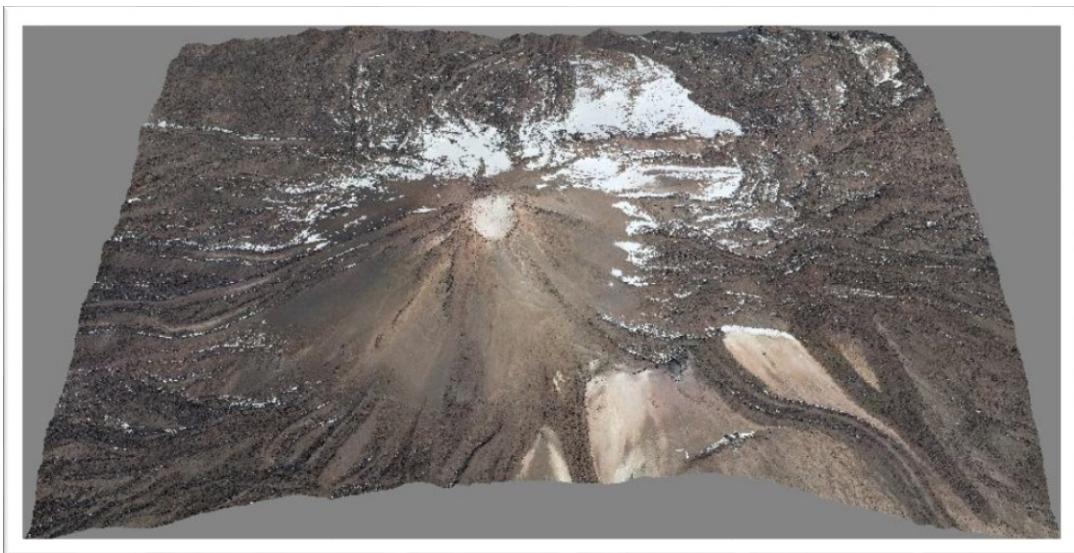


Figura 10.34. Modelo tridimensional de nube de puntos del edificio del Teide, generado con el software MEL, con color RGB a partir del archivo PNOA_2016_CANAR-TF_338-3130_ORT-CLA-RGB.laz del IGN
Fuente: elaboración propia, a partir de datos del IGN

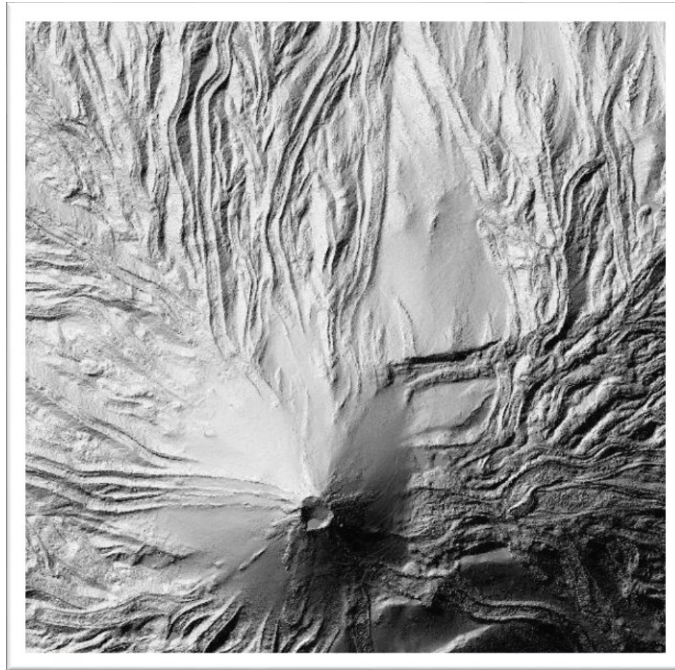


Figura 10.35. Hillshade del edificio del Teide con resolución de malla de 25 cm, generado con el software MEL a partir del archivo PNOA_2016_CANAR- TF_338-3130_ORT-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN

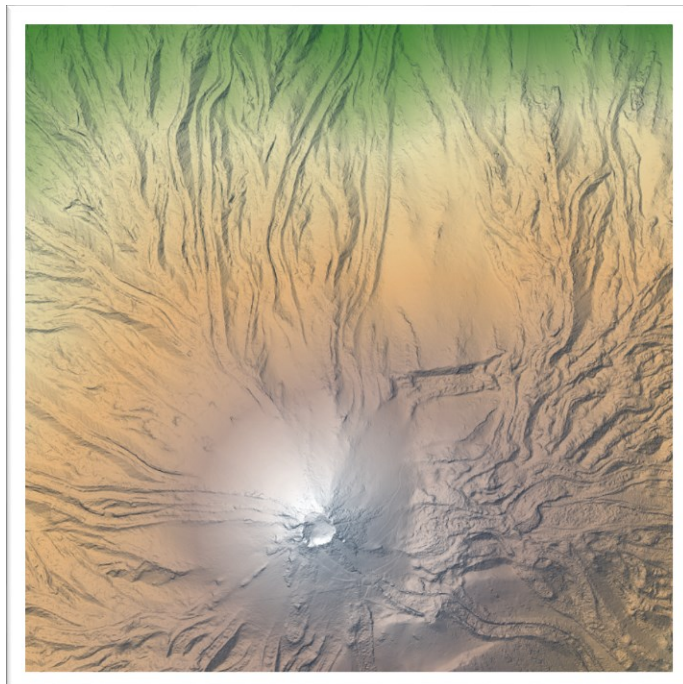


Figura 10.36. Mapa hipsométrico del edificio del Teide con resolución de malla de 25 cm, generado con el software MEL a partir del archivo PNOA_2016_CANAR- TF_338-3130_ORT-CLA-RGB.laz del IGN (paleta: high_relief)

Fuente: elaboración propia, a partir de datos del IGN

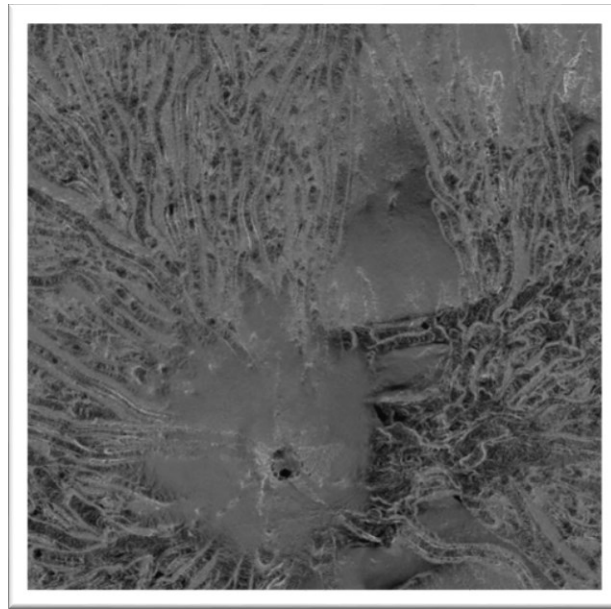


Figura 10.37. Mapa de pendientes del edificio del Teide con resolución de malla de 25 cm, generado con el software MEL a partir del archivo PNOA_2016_CANAR-TF_338-3130_ORT-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN

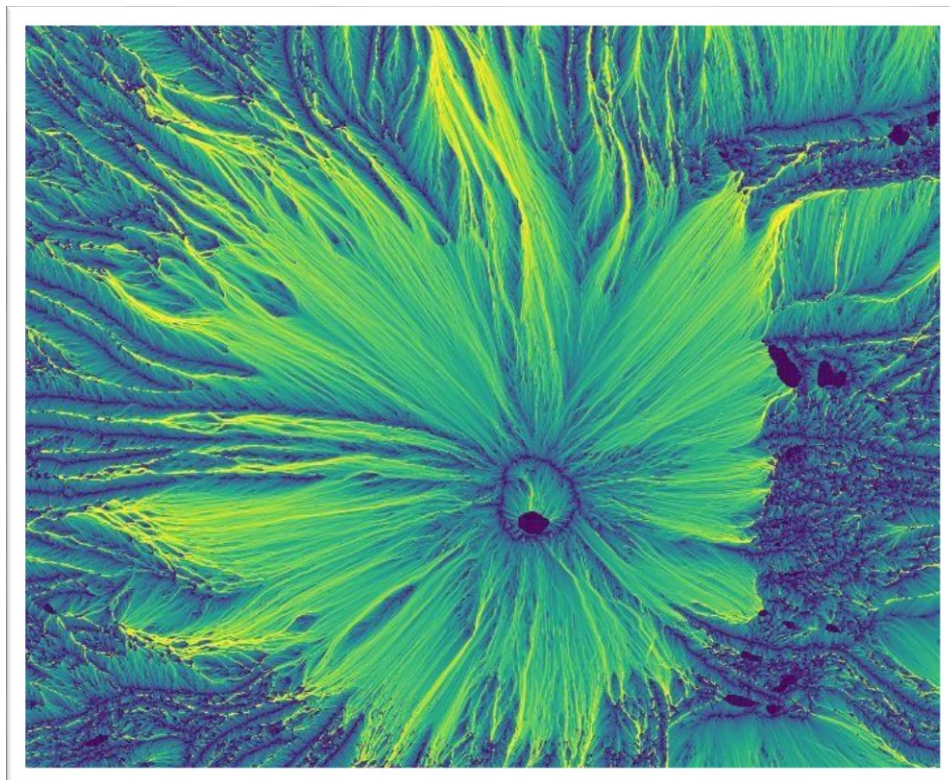


Figura 10.38. Detalle del edificio del Teide en el mapa de acumulación de flujo con resolución de malla de 25 cm y pseudocolor monobanda, generado con el software MEL a partir del archivo PNOA_2016_CANAR-TF_338- 3130_ORT-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN

10.6. Edificios de la pared de Las Cañadas (Tenerife)

Nombre de los archivos LiDAR utilizados	PNOA_2016_CANAR-TF_342-3124_ORT-CLA-RGB.laz PNOA_2016_CANAR-TF_344-3124_ORT-CLA-RGB.laz
X (UTM ETRS89):	344369
Y (UTM ETRS89)	3122177
Huso	28

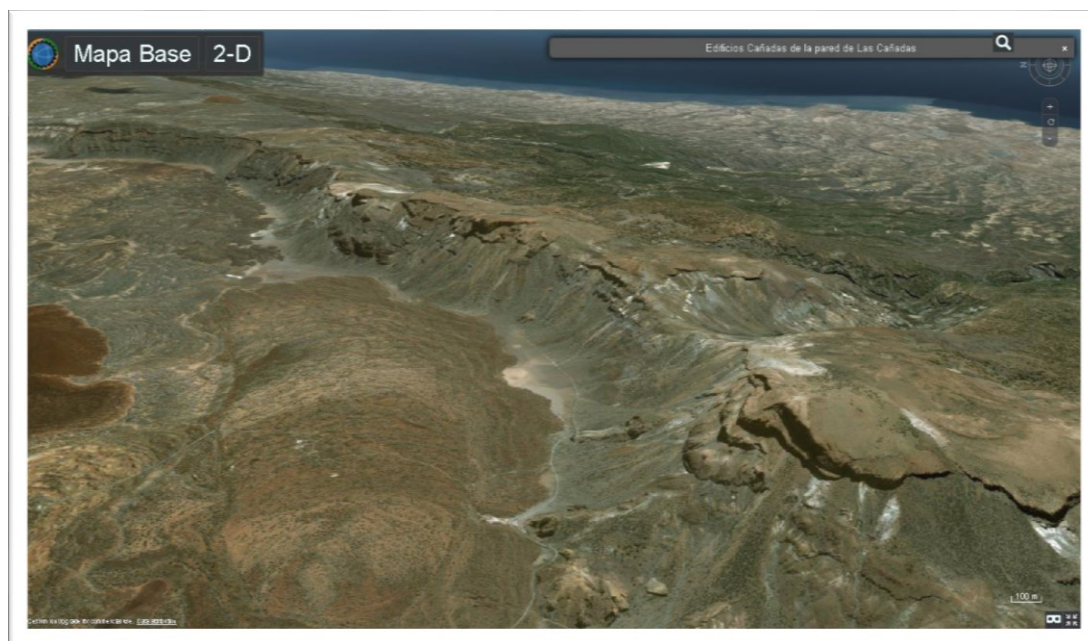


Figura 10.39. Modelo tridimensional de los edificios de la pared de Las Cañadas, generado con el software online para el estudio del relieve, con texturas de Microsoft Satélite
Fuente: elaboración propia, a partir de datos de Microsoft, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia

Las imágenes incluidas en este ejemplo muestran la pared de Las Cañadas, donde puede apreciarse la superposición de edificios en el espacio y tiempo, debidos a la sucesión volcánica producida en esta zona. Las imágenes generadas con el software desarrollado permiten contemplar los pronunciados

escarpes existentes en la pared de la gran caldera volcánica de Las Cañadas, en estos escarpes es posible ver la sucesión volcanoestratigráfica de coladas y rocas piroclásticas; así mismo, los modelos tridimensionales, a los que se han incluido como texturas las ortoimágenes del PNOA y de la capa Microsoft Satélite muestran claramente depósitos sedimentarios.

PNOA_2016_CANAR-TF_342-3124_ORT-CLA-RGB.laz

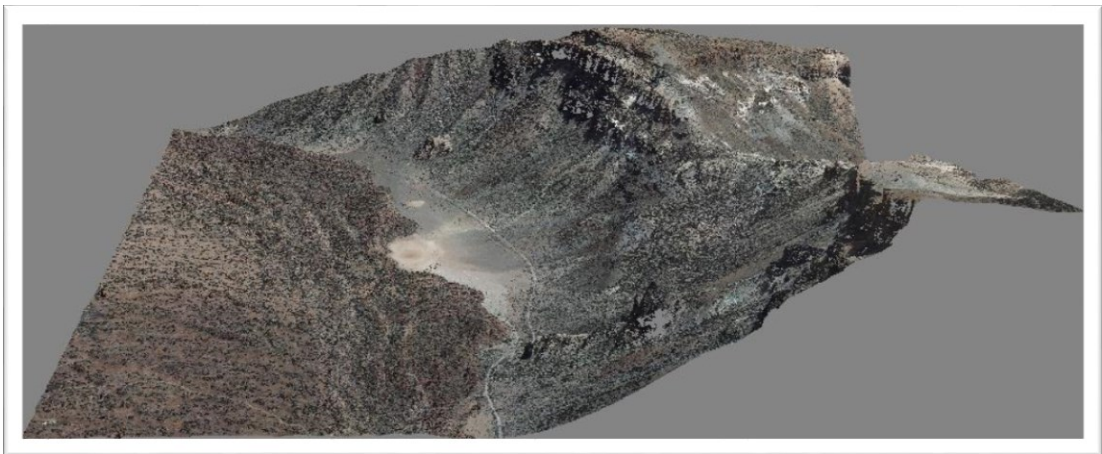


Figura 10.40. Modelo tridimensional de nube de puntos de los edificios de la pared de Las Cañadas, generado con el software MEL, con color RGB, a partir del archivo PNOA_2016_CANAR-TF_342-3124_ORT-CLA-RGB.laz del IGN
Fuente: elaboración propia, a partir de datos del IGN

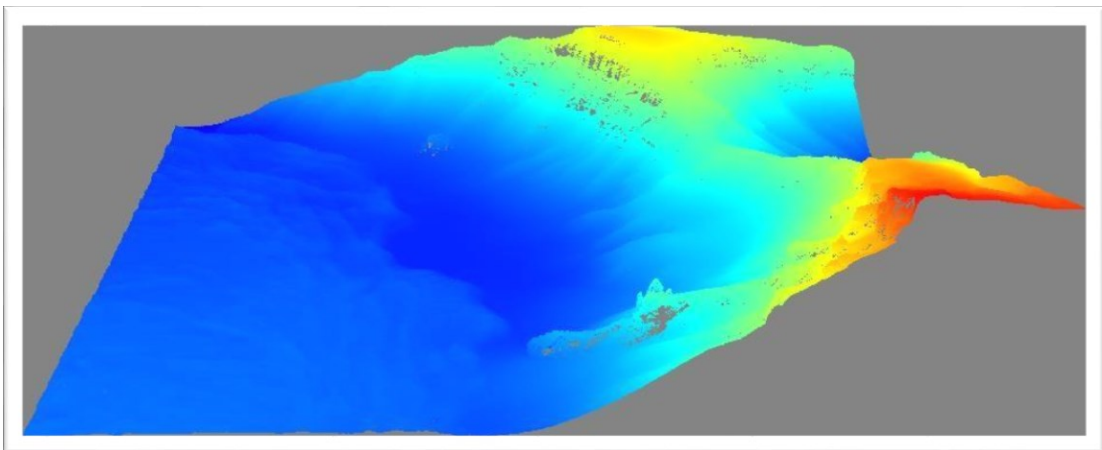


Figura 10.41. Modelo tridimensional de nube de puntos de los edificios de la pared de Las Cañadas, generado con el software MEL, con color en función de la altura, a partir del archivo PNOA_2016_CANAR-TF_342-3124_ORT-CLA-RGB.laz del IGN
Fuente: elaboración propia, a partir de datos del IGN

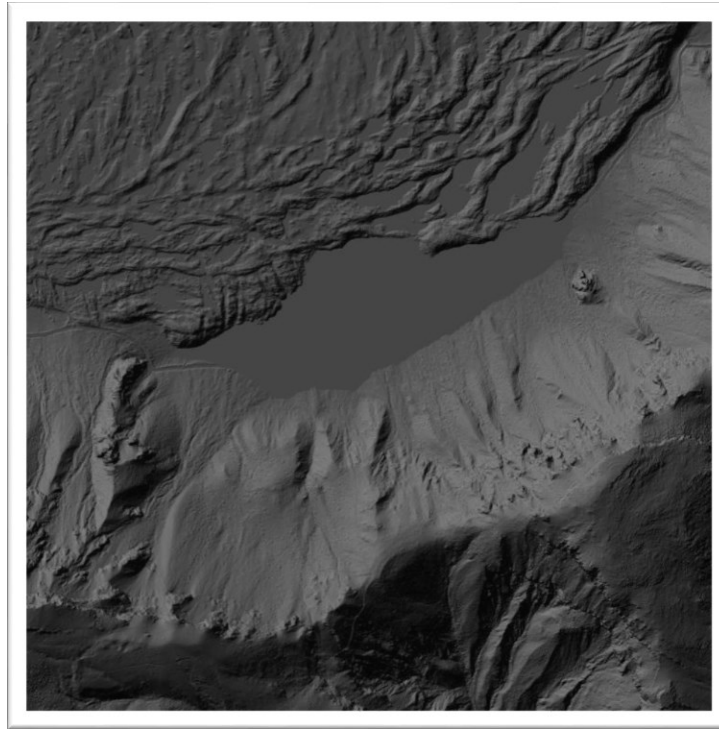


Figura 10.42. Hillshade de los edificios de la pared de Las Cañadas con resolución de malla de 25cm, generado con el software MEL a partir del archivo PNOA_2016_CANAR-TF_342-3124_ORI-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN

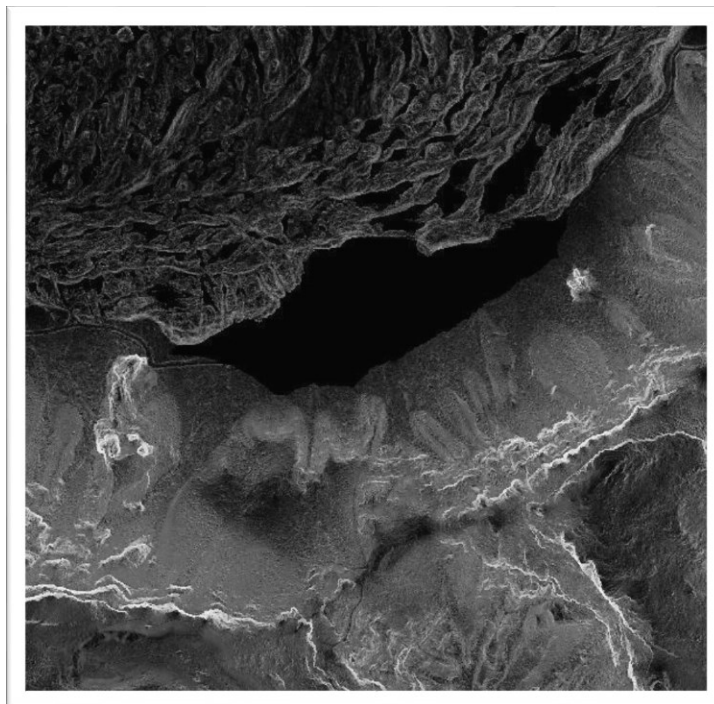


Figura 10.43. Mapa de pendientes de los edificios de la pared de Las Cañadas con resolución de malla de 25 cm, generado con el software MEL a partir del archivo PNOA_2016_CANAR-TF_342-3124_ORI-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN

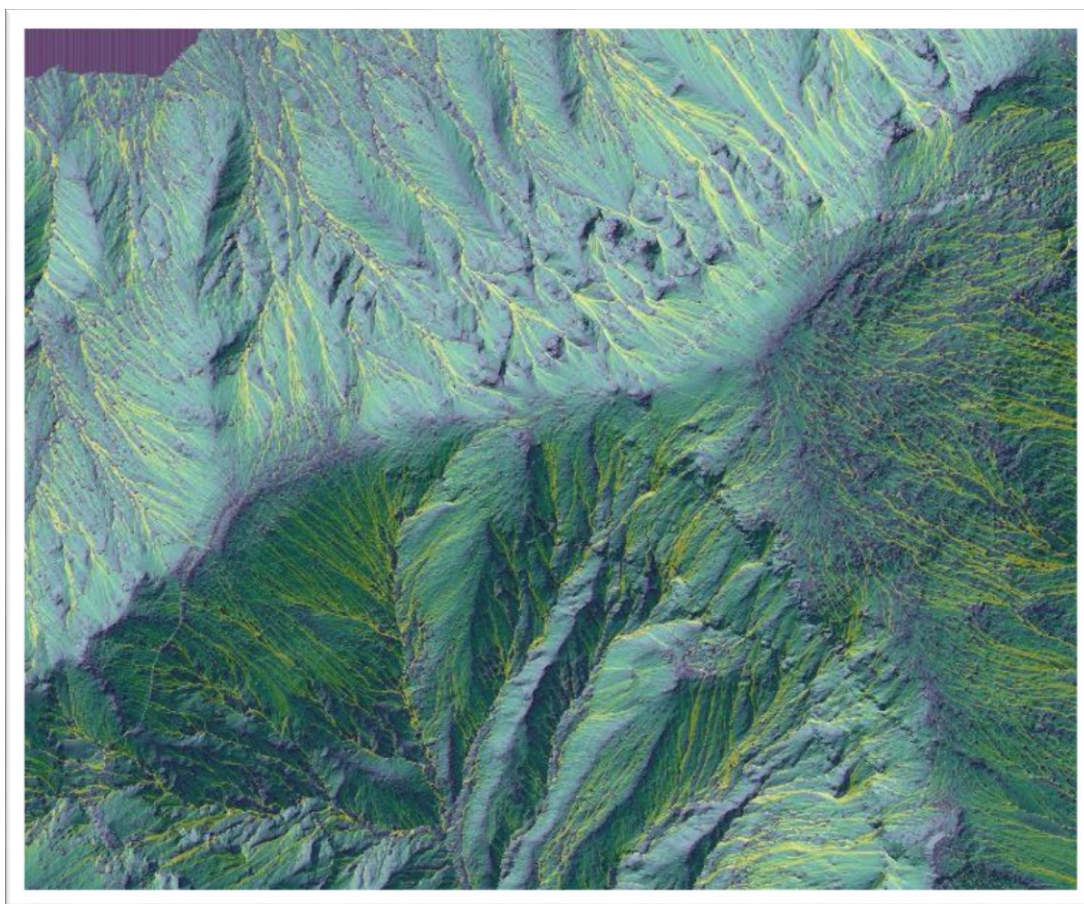


Figura 10.44. Combinación del mapa de acumulación de flujo con pseudocolor monobanda y el hillshade de los edificios de la pared de Las Cañadas con resolución de malla de 25 cm, generado con el software MEL a partir del archivo PNOA_2016_CANAR-TF_342-3124_ORT-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN

PNOA 2016 CANAR-TF 344-3124 ORT-CLA-RGB.laz

En las visualizaciones de los datos generados a partir de este archivo LiDAR pueden observarse formaciones geológicas como el Roque de la Grieta (esquina superior izquierda) y la Montaña de Palo (centro derecha).

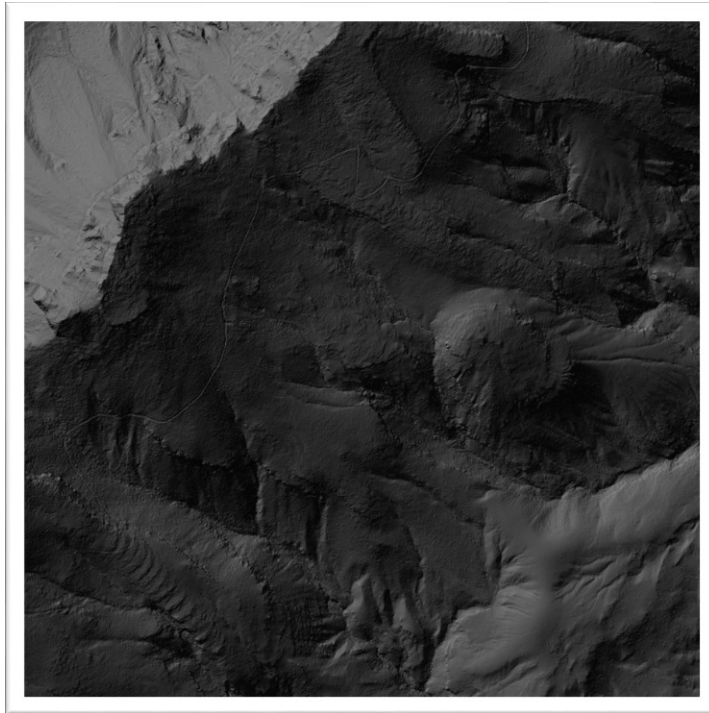


Figura 10.45. Hillshade de los edificios de la pared de Las Cañadas con resolución de malla de 25cm, generado con el software MEL a partir del archivo PNOA_2016_CANAR-TF_344-3124_ORT-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN

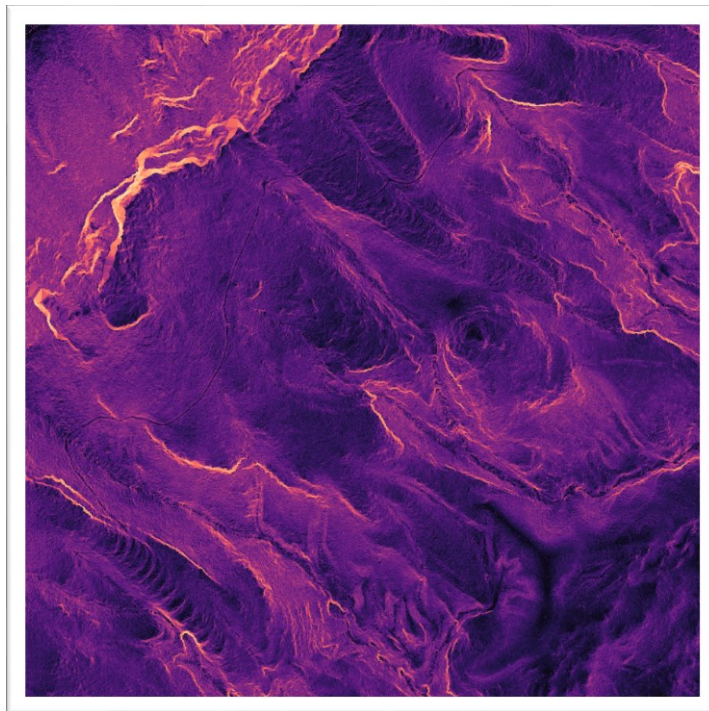


Figura 10.46. Mapa de pendientes con pseudocolor monobanda de los edificios de la pared de Las Cañadas con resolución de malla de 25 cm, generado con el software MEL a partir del archivo PNOA_2016_CANAR-TF_344-3124_ORT-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN

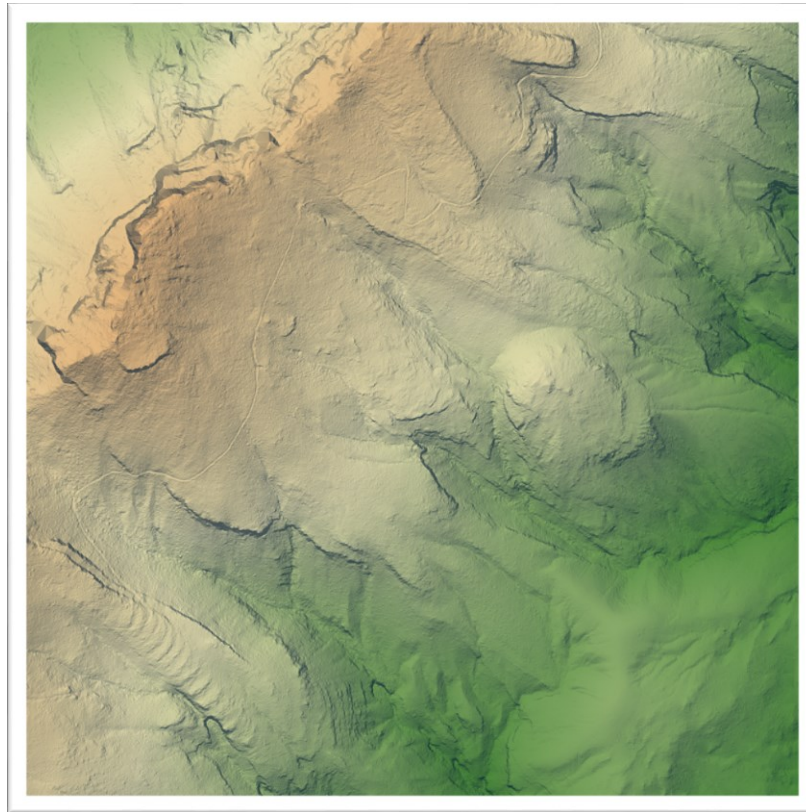


Figura 10.47. Mapa hipsométrico de los edificios de la pared de Las Cañadas con resolución de malla de 25 cm, generado con el software MEL a partir del archivo PNOA_2016_CANAR-TF_344-3124_ORT-CLA-RGB.laz del IGN
(paleta: atlas)

Fuente: elaboración propia, a partir de datos del IGN

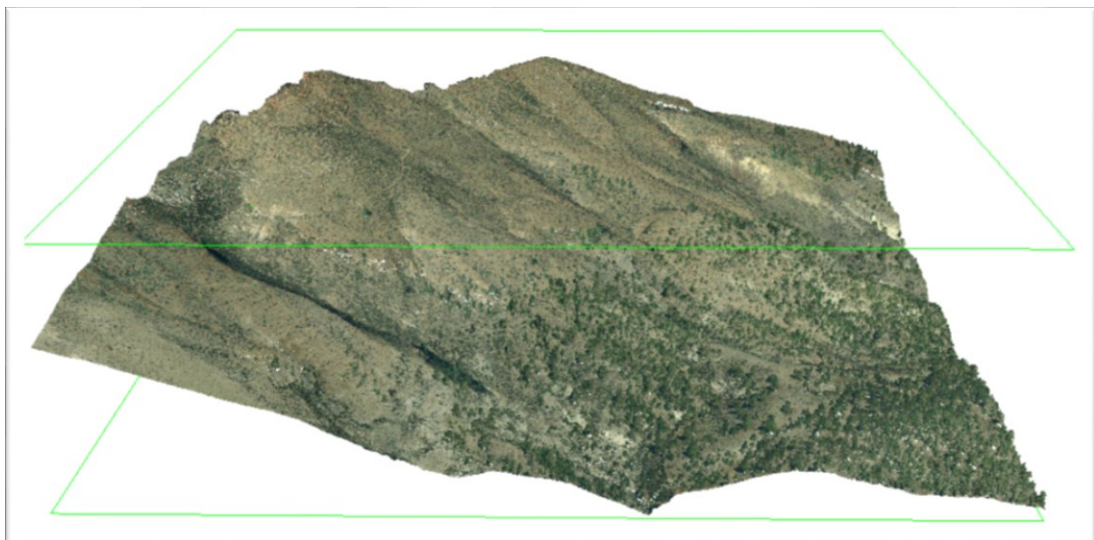


Figura 10.48. Modelo tridimensional de nube de puntos de los edificios de la pared de Las Cañadas, generado con el software MEL, con color RGB, a partir del archivo PNOA_2016_CANAR-TF_344-3124_ORT-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN

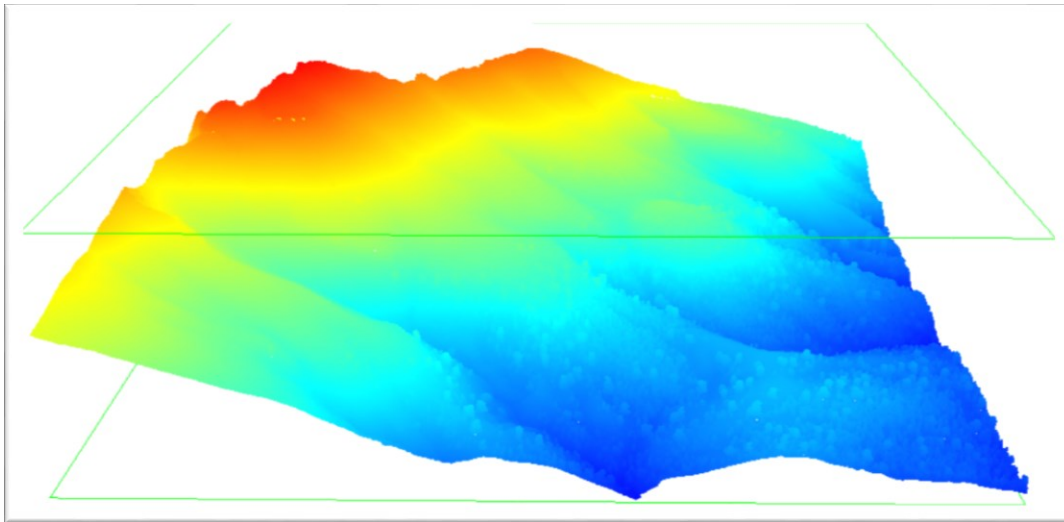


Figura 10.49. Modelo tridimensional de nube de puntos de los edificios de la pared de Las Cañadas, generado con el software MEL, con color en función de la altura, a partir del archivo PNOA_2016_CANAR-TF_344- 3124_ORT-CLA-RGB.laz del IGN
Fuente: elaboración propia, a partir de datos del IGN

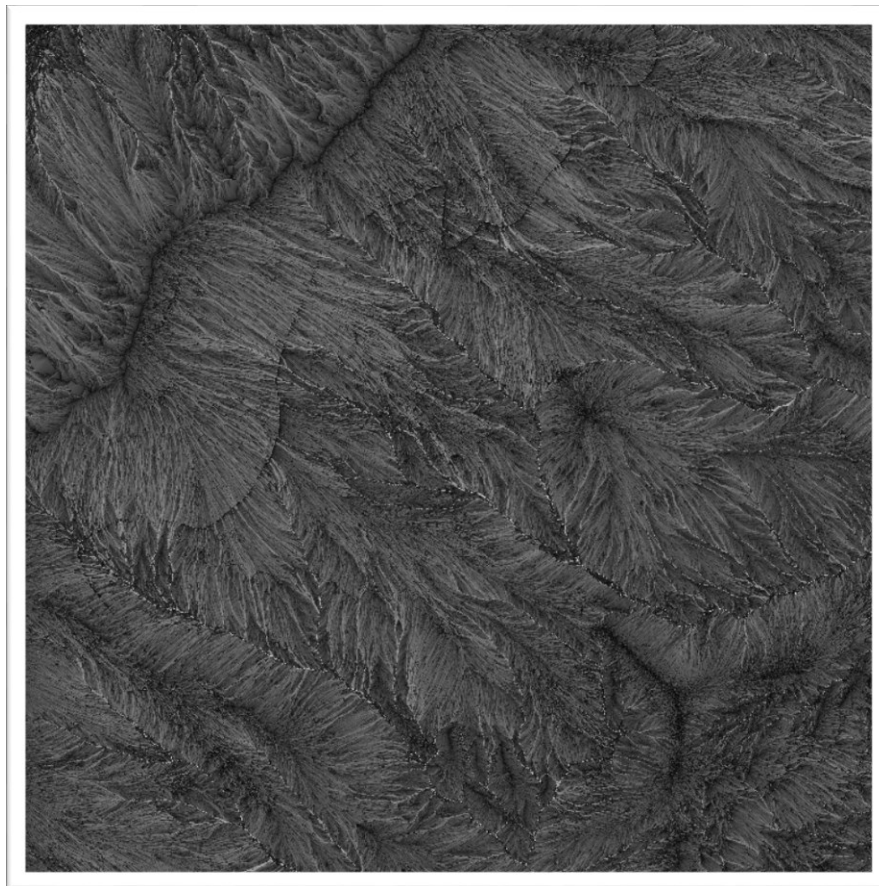


Figura 10.50. Mapa de pendientes de los edificios de la pared de Las Cañadas con resolución de malla de 25 cm y pseudocolor monobanda, generado con el software MEL a partir del archivo PNOA_2016_CANAR-TF_344- 3124_ORT-CLA-RGB.laz del IGN
Fuente: elaboración propia, a partir de datos del IGN

10.7. Roque Nublo (Tenerife)

Nombre del archivo LiDAR utilizado	PNOA_2015_CANAR-GC_438-3096_ORT-CLA-RGB.laz
X (UTM ETRS89)	439761
Y (UTM ETRS89)	3094122
Huso	28

En este ejemplo pueden apreciarse los depósitos de avalancha del Parque Rural del Nublo, resultado del colapso sectorial de un estratovolcán en el centro de Gran Canaria. Es especialmente llamativo el Monolito conocido como Roque Nublo, al cual la legislación ha singularizado como Monumento Natural. Debido a la gran resolución obtenida con el software MEL, es posible apreciar, no sólo el Roque Nublo, sino también la vegetación que lo arropa compuesta por pino canario de repoblación que convive con el matorral de retamas, codesos y salvias. Los modelos tridimensionales, el mapa hipsométrico, el hillshade y, sobre todo, los mapas de pendientes nos permiten apreciar los fuertes escarpes y las estructuras desmanteladas existentes en el área analizada.

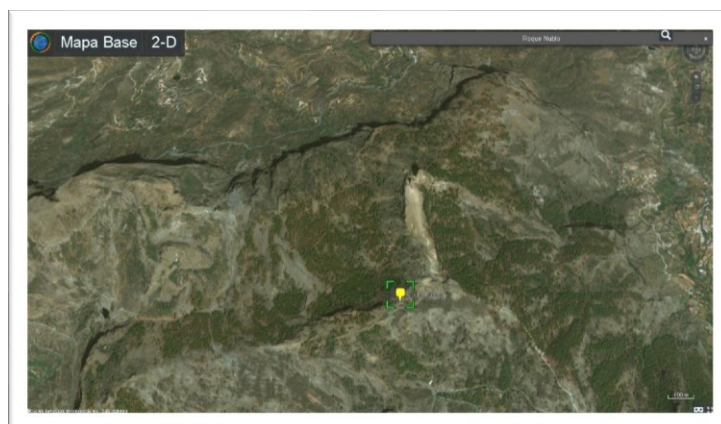


Figura 10.51. Localización del Roque Nublo, imagen generada con el software online para el estudio del relieve, con texturas de Microsoft Satélite

Fuente: elaboración propia, a partir de datos del IGN, el IGME, el Servicio Geológico de los Estados Unidos, CGIAR-CSI, Copernicus, Land Information New Zealand, data.gov.uk y Geoscience Australia



Figura 10.52. Detalle del Roque Nublo en el modelo tridimensional de nube de puntos con color RGB, generado con el software MEL a partir del archivo PNOA_2015_CANAR-GC_438-3096_ORT-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN



Figura 10.53. Hillshade generado con el software MEL a partir del archivo PNOA_2015_CANAR-GC_438-3096_ORT-CLA-RGB.laz del IGN

Fuente: elaboración propia, a partir de datos del IGN



Figura 10.54. Mapa hipsométrico generado con el software MEL a partir del archivo PNOA_2015_CANAR-GC_438-3096_ORT-CLA-RGB.laz del IGN (paleta: atlas)
Fuente: elaboración propia, a partir de datos del IGN

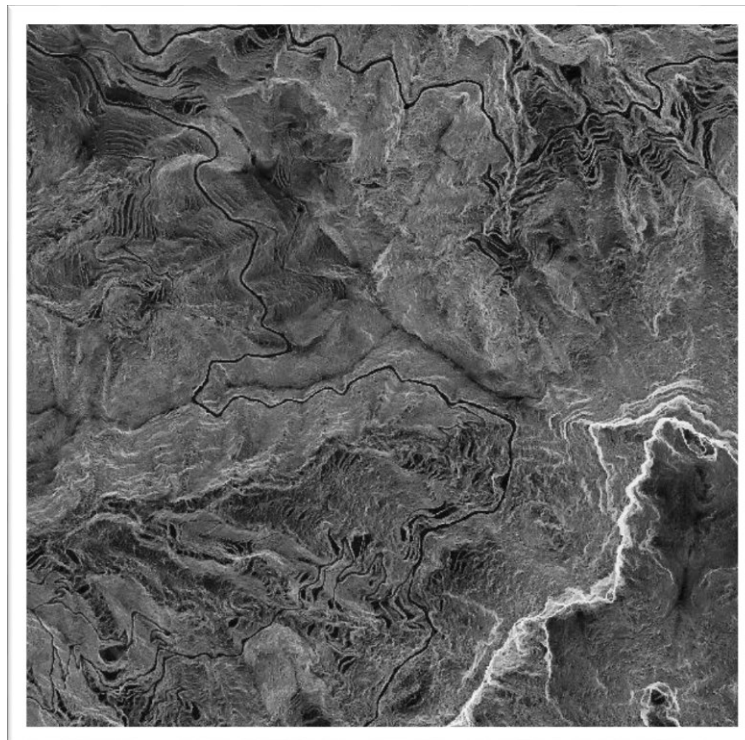


Figura 10.55. Mapa de pendientes generado con el software MEL a partir del archivo PNOA_2015_CANAR-GC_438-3096_ORT-CLA-RGB.laz del IGN
Fuente: elaboración propia, a partir de datos del IGN

10.8. Aplicaciones geoarqueológicas

Entre la diversidad de utilidades que ofrecen las herramientas creadas en esta tesis, una de las más llamativas es la posibilidad de ser utilizadas para la teledetección de posibles estructuras con importancia arqueológica. Gracias a que la tecnología LiDAR es capaz de atravesar la vegetación y categorizar los puntos de las nubes de puntos tridimensionales, podemos eliminar la vegetación e identificar estructuras que permanecen ocultas, las cuales no serían identificables mediante los métodos tradicionales de teledetección basados en ortofotografía. De esta forma, esta tecnología se reafirma como un instrumento imprescindible en la investigación arqueológica.

Al objeto de ilustrar lo anteriormente mencionado, seguidamente se expone el resultado de la aplicación práctica de esta tecnología sobre el territorio nacional.

Durante el desarrollo de esta herramienta, se decidió hacer las pruebas de esta sobre la zona del noroeste peninsular, concretamente en la comunidad autónoma de Galicia, dadas las características orográficas que allí se dan, así como la humedad habitual de esa zona, que propician la abundante vegetación dominante.

En dichas pruebas, se identificaron diferentes formas, de origen aparentemente antrópico, pues estaban muy definidas, lo que motivó buscar una explicación para las mismas. Una de las figuras obtenidas tenía una similitud sorprendente con los característicos castros celtas de esa zona, lo cual derivó, asimismo, en el interés por esa rama de la arqueología. El resultado de estas pruebas, por tanto, fue la adquisición de unos conocimientos elementales sobre estas antiguas construcciones, ya que tenían unas características comunes en todos los casos, coincidiendo con las encontradas en las pruebas efectuadas con la aplicación MEL.

A fin de facilitar la comprensión del ejemplo aquí expuesto y de las imágenes aportadas, se considera preciso dar unas nociones básicas sobre la cultura castreña, en base a lo averiguado por los arqueólogos hasta la actualidad.

El elemento fundamental de dicha cultura, evidentemente, son los castros, que eran pequeños poblados levantados sobre colinas y fortificados con grandes murallas, disponiendo en ocasiones también de fosos que rodeaban el perímetro externo. Otro rasgo característico de los castros era su morfología circular u oval, tanto en la fortificación como en las viviendas.

Dentro de estas fortificaciones se encontraban las viviendas familiares, siendo toda la construcción de mampostería. Desde su emplazamiento elevado controlaban las tierras y/o playas que se extendían frente a ellos.

Estos núcleos de población se encontraban diseminados por el citado territorio, y estaban cercanos entre sí, pero la situación del poblado era la que marcaba el entorno a explotar por cada uno de ellos.

La razón de que este tipo de poblados tuvieran un carácter defensivo se debería a que no estaban unidos, por lo que debían ser frecuentes las rivalidades entre comunidades vecinas.

Tal como se indicaba en el inicio de este epígrafe y una vez realizada esta breve introducción para una mejor comprensión de las imágenes presentadas, así como del contexto en el que se produjeron las construcciones antrópicas en ellas contenidas, seguidamente se expone un ejemplo, resultado de la aplicación llevada a cabo mediante la tecnología desarrollada en este trabajo en el campo de la arqueología. Para ello, se parte de una fotografía de uno de los más importantes castros celtas descubiertos y estudiados en la Comunidad Autónoma de Galicia, el castro de Viladonga, a fin de que sirva de referente a la

hora de comparar los paralelismos con las imágenes de las pruebas aquí realizadas.



Figura 10.56. Fotografía del Castro de Viladonga.

Fuente: <https://twitter.com/viajandopiedras/status/1033498330289123335>

10.8.1. Lugar Porto Outeiro

Tipología	Posible Castro Celta
Época	Edad del Hierro
Ubicación	<p>País: España</p> <p>Comunidad Autónoma: Galicia</p> <p>Provincia: Lugo</p> <p>Localidad: Porto Outeiro</p> <p>Lugar: Porto</p> <p>Coordenadas:</p> <p>Latitud: 43° 9' 39.38" N</p> <p>Longitud: 8° 3' 22.77" O</p>
Rasgos tipológicos	<p>Superficie: 1,37 Has.</p> <p>Forma: Circular</p>
Nombre del archivo LiDAR utilizado	PNOA_2010_Lote10_GAL_576-4780_ORT-CLA-COL.laz
Localización cartográfica	<p>La localización que se muestra a continuación ha sido obtenida mediante la generación de un KML a partir de la nube de puntos contenida en el archivo PNOA_2010_Lote10_GAL_576-4780_ORT-CLA-COL.laz, mediante el software "Morillas' Easy LiDAR", tal y como se muestra en el capítulo "3.2. Desarrollo del Software de escritorio MEL del Bloque I de este trabajo.</p>



Figura 10.57. Imagen de la cartografía del IGN, en rojo la localización. EPSG: 25829
Fuente: elaboración propia, a partir de datos del IGN



Figura 10.58. Detalle de la extensión del archivo de nube de puntos LiDAR utilizado, sobre cartografía del IGN EPSG: 25829
Fuente: elaboración propia, a partir de datos del IGN



Figura 10.59. Detalle de la extensión del archivo de nube de puntos LiDAR utilizado sobre imagen del PNOA de Máxima Actualidad
Fuente: elaboración propia, a partir de datos del IGN



Figura 10.60. Imagen del PNOA, en rojo la localización. EPSG: 25829
Fuente: elaboración propia, a partir de datos del IGN

Al filtrar las imágenes en función de los factores que queramos ver o eliminar, nos descubre una realidad oculta y sorprendente, donde cambia la superficie apreciable. Esto es lo que ocurre en el caso que se está tratando, como se puede ver en las fotografías abajo insertadas, donde, mediante el filtrado de los datos LiDAR, se puede visualizar una construcción con forma idéntica a la de los castros habituales de la zona, de los cuales el ya citado castro de Viladonga es un claro ejemplo, que ya está siendo excavado y estudiado (se ha adjuntado imagen anteriormente). Si comparamos la imagen derivada de los datos LiDAR, filtrados con el software online para el estudio del relieve, se puede comprobar perfectamente cómo, a pesar de tratarse de las mismas coordenadas, en la fotografía no hay constancia de la estructura al estar la superficie cubierta por la vegetación.



Figura 10.61. Detalle de vista tridimensional del lugar donde se encuentra el posible castro en Porto Outeiro cubierto completamente de vegetación, generada con el software online, utilizando la capa Microsoft Satélite.

Fuente: elaboración propia, a partir de datos del IGN



Figura 10.62. Modelo tridimensional de nube de puntos sin filtrado de los datos, con color RGB

Fuente: elaboración propia, a partir de datos del IGN

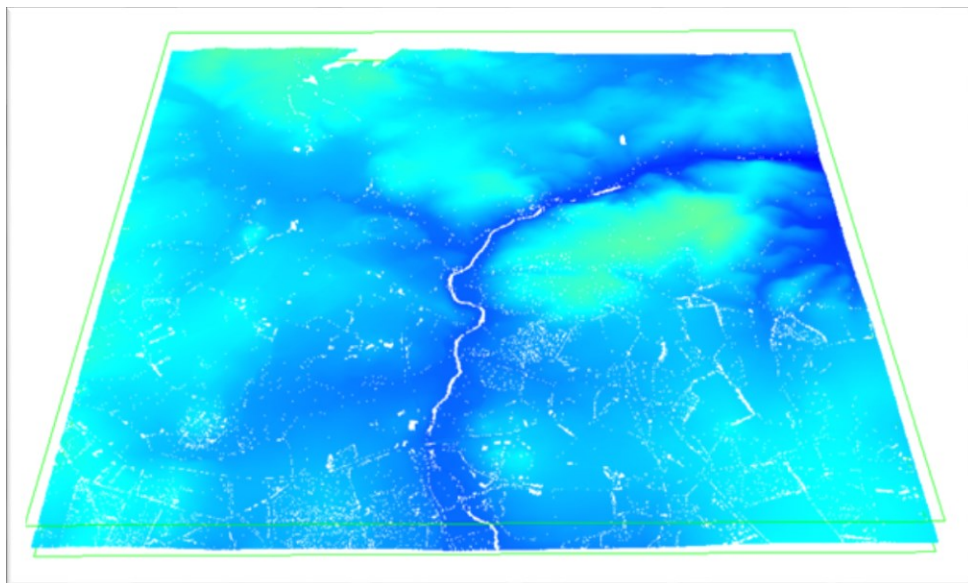


Figura 10.63. Modelo tridimensional de nube de puntos filtrando los datos para mostrar sólo el terreno y con coloración según la altura

Fuente: elaboración propia, a partir de datos del IGN

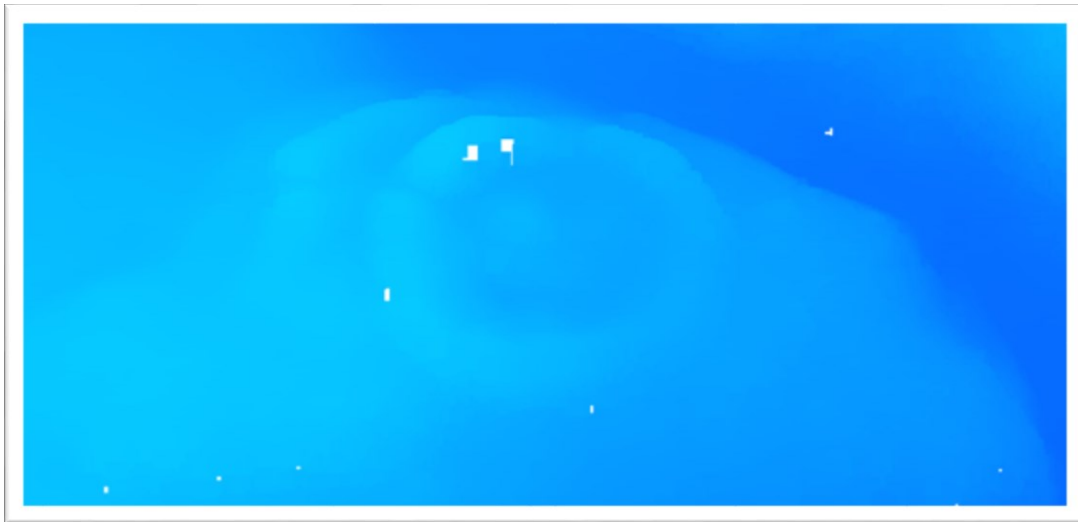


Figura 10.64. Detalle del posible castro en modelo tridimensional de nube de puntos con magnificación de la altura y coloración según altura.
Fuente: elaboración propia, a partir de datos del IGN

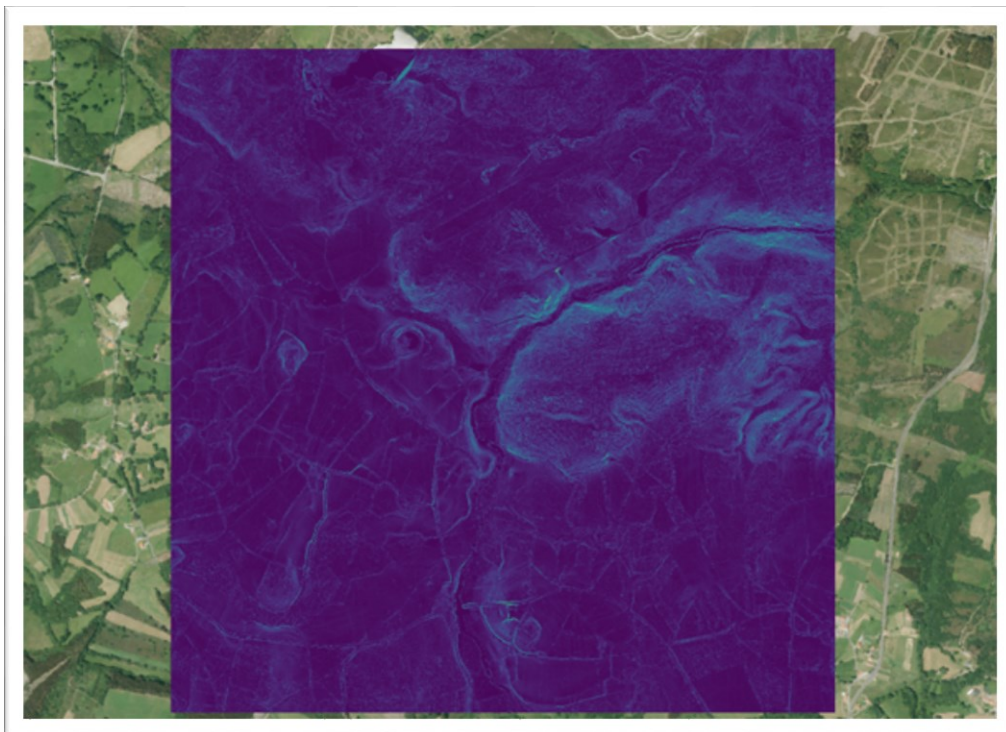


Figura 10.65. Mapa de pendientes con pseudocolor monobanda, de resolución de malla de 25 cm obtenida con el software MEL sobre mosaico de ortoimágenes del PNOA de máxima actualidad. EPSG: 25829.
Fuente: elaboración propia, a partir de datos del IGN

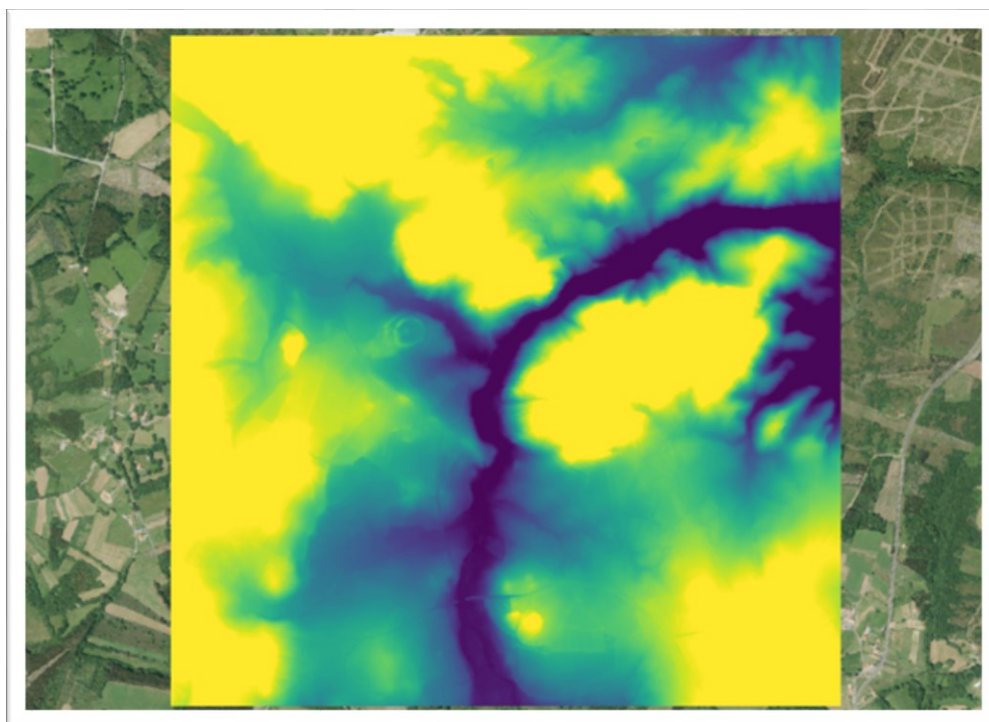


Figura 10.66. MDE con pseudocolor monobanda, de resolución de malla de 25 cm obtenida con el software MEL sobre mosaico de ortoimágenes del PNOA de Máxima Actualidad. EPSG: 25829.

Fuente: elaboración propia, a partir de datos del IGN

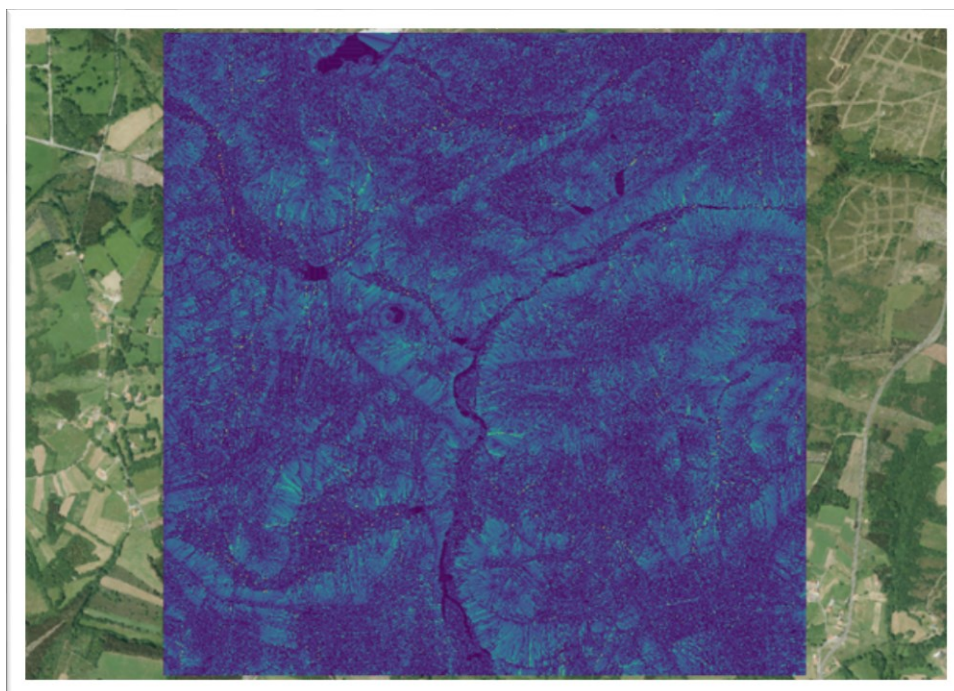


Figura 10.67. Mapa de acumulación de flujo, con pseudocolor monobanda, de resolución de malla de 25 cm obtenida con el software MEL sobre mosaico de ortoimágenes del PNOA de Máxima Actualidad. EPSG: 25829

Fuente: elaboración propia, a partir de datos del IGN

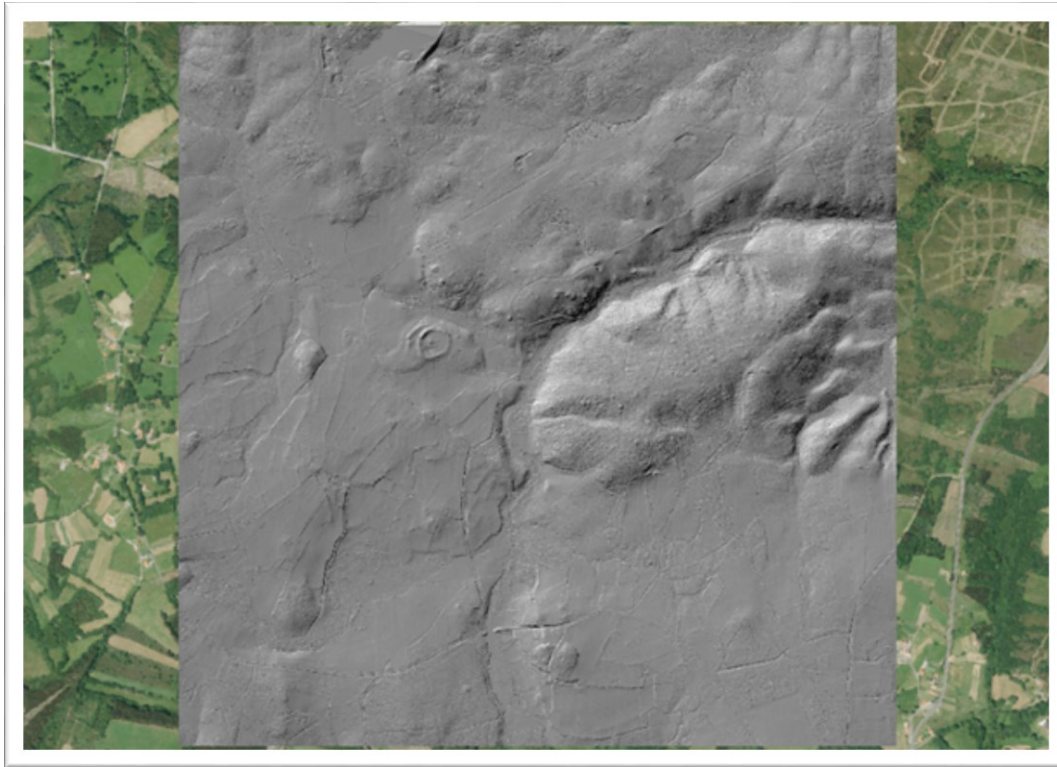


Figura 10.68. Hillshade de resolución de malla de 25 cm obtenida con el software MEL sobre mosaico de ortoimágenes del PNOA de Máxima Actualidad. EPSG: 25829
Fuente: elaboración propia, a partir de datos del IGN



Figura 10.69. Modelo tridimensional generado a partir del MDE producido con el software MEL con resolución de malla de 25 cm. EPSG: 25829
Fuente: elaboración propia, a partir de datos del IGN

En la imagen anterior, y una vez eliminados los datos correspondientes a vegetación y generando un modelo tridimensional, se puede distinguir perfectamente la típica silueta de los castros gallegos, incluyendo el detalle de las entradas, lo que sería indicativo de una buena conservación.



Figura 10.70. Vista tridimensional de la nube de puntos con coloración RGB correspondiente al PNOA, en la que no se han aplicado los filtros para eliminar la vegetación, lo que impide distinguir cualquier tipo de estructura

Fuente: elaboración propia, a partir de datos del IGN

A continuación, la figura 10.71 muestra una panorámica mayor del modelo tridimensional generado a partir del filtrado de datos LiDAR del mismo enclave recogido en la figura 10.70.



Figura 10.71. Modelo tridimensional generado a partir del MDE generado con el software MEL con resolución de malla de 25 cm y filtrado de categorías para representar únicamente el terreno, al que se le han aplicado diversos filtros de imagen para mejorar la identificación de estructuras. EPSG: 25829

Fuente: elaboración propia, a partir de datos del IGN



Figura 10.72. Detalle de ortoimagen del PNOA de Máxima Actualidad con la misma extensión que las imágenes mostradas a continuación. EPSG: 25829.

Fuente: elaboración propia, a partir de datos del IGN

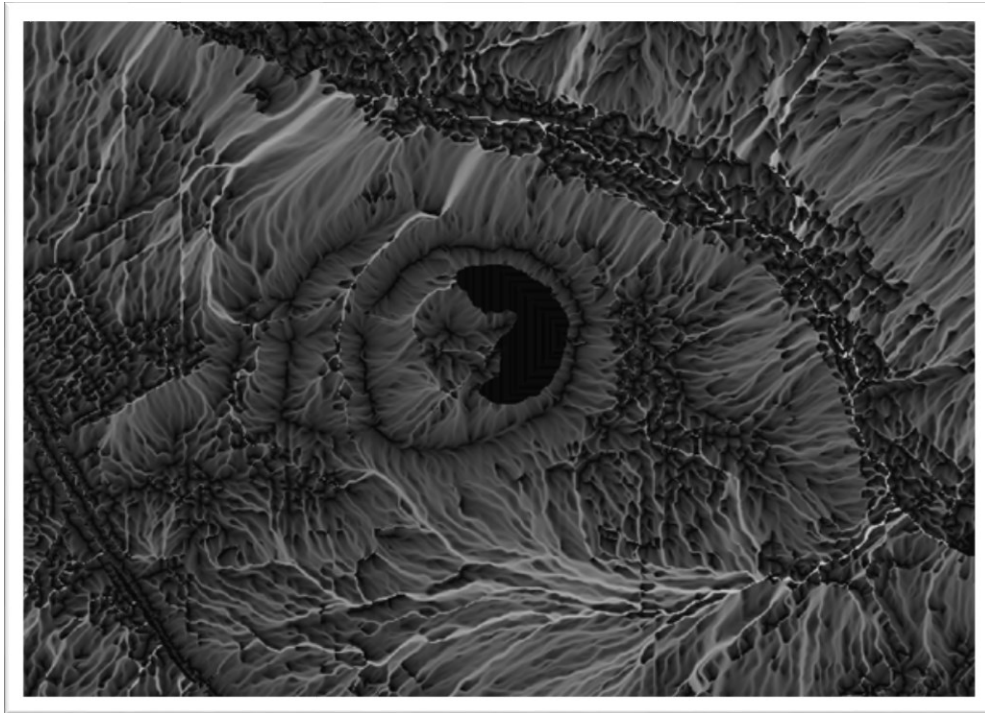


Figura 10.73. Detalle del mapa de acumulación de flujo generado con el software MEL con resolución de malla de 25 cm. EPSG: 25829
Fuente: elaboración propia, a partir de datos del IGN

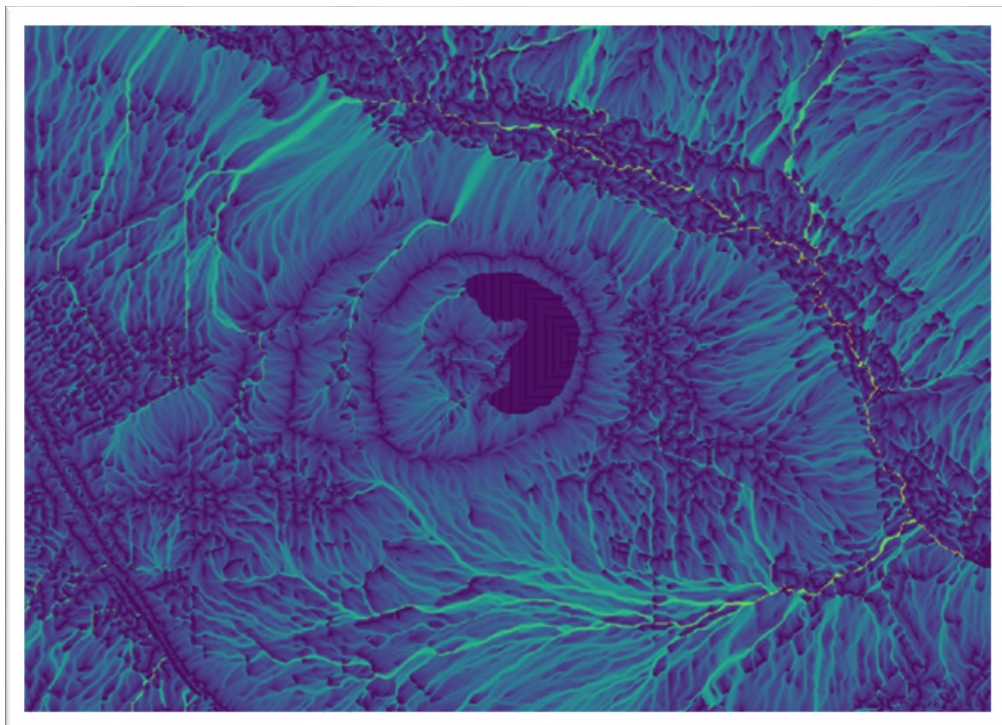


Figura 10.74. Detalle del mapa de acumulación de flujo generado con el software MEL con resolución de malla de 25 cm y coloreado con pseudocolor monobanda. EPSG: 25829
Fuente: elaboración propia, a partir de datos del IGN

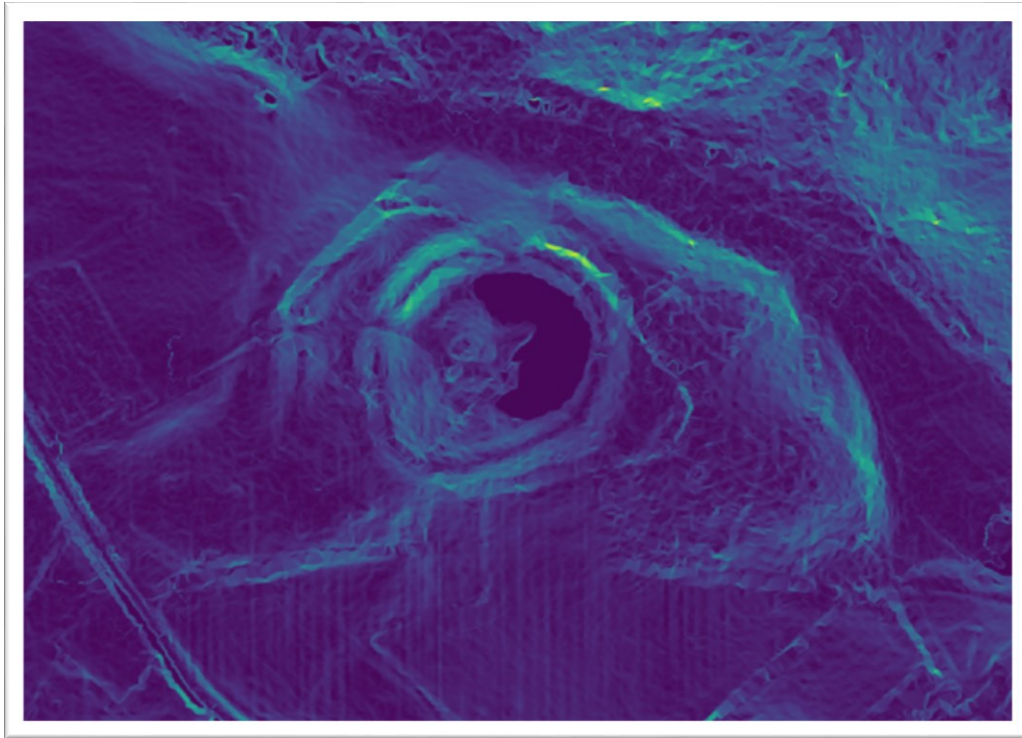


Figura 10.75. Detalle del mapa de pendientes generado con el software MEL con resolución de malla de 25 cm y coloreado con pseudocolor monobanda. EPSG: 25829.
Fuente: elaboración propia, a partir de datos del IGN

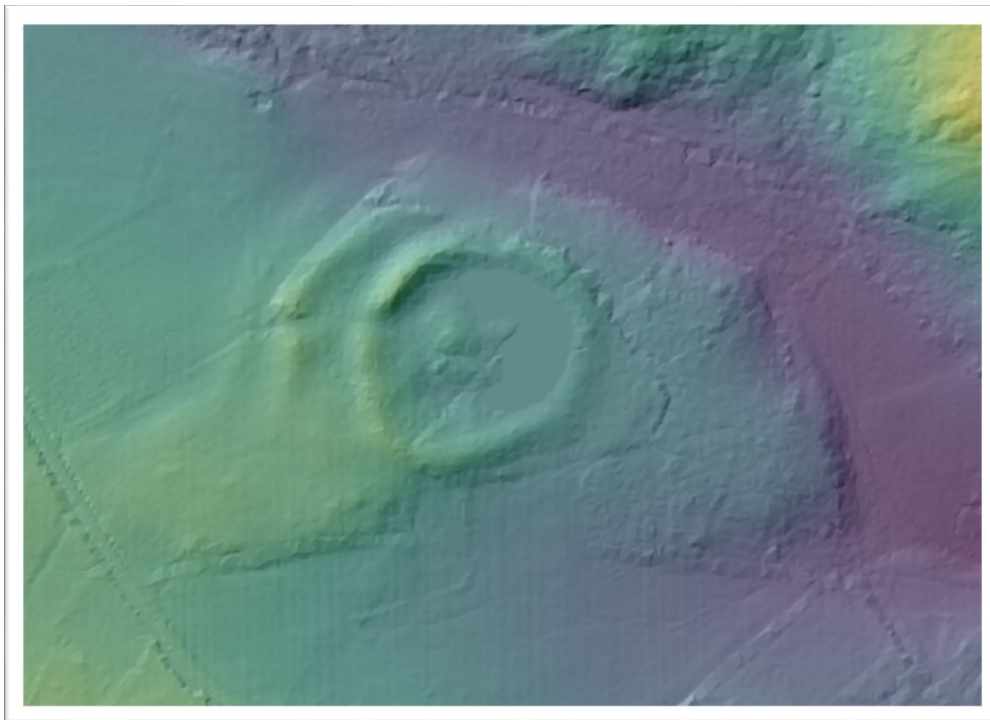


Figura 10.76. Detalle del hillshade combinado con el MDE con pseudocolor monobanda, ambas capas generadas con el software MEL, con resolución de malla de 25 cm. EPSG: 25829

Fuente: elaboración propia, a partir de datos del IGN

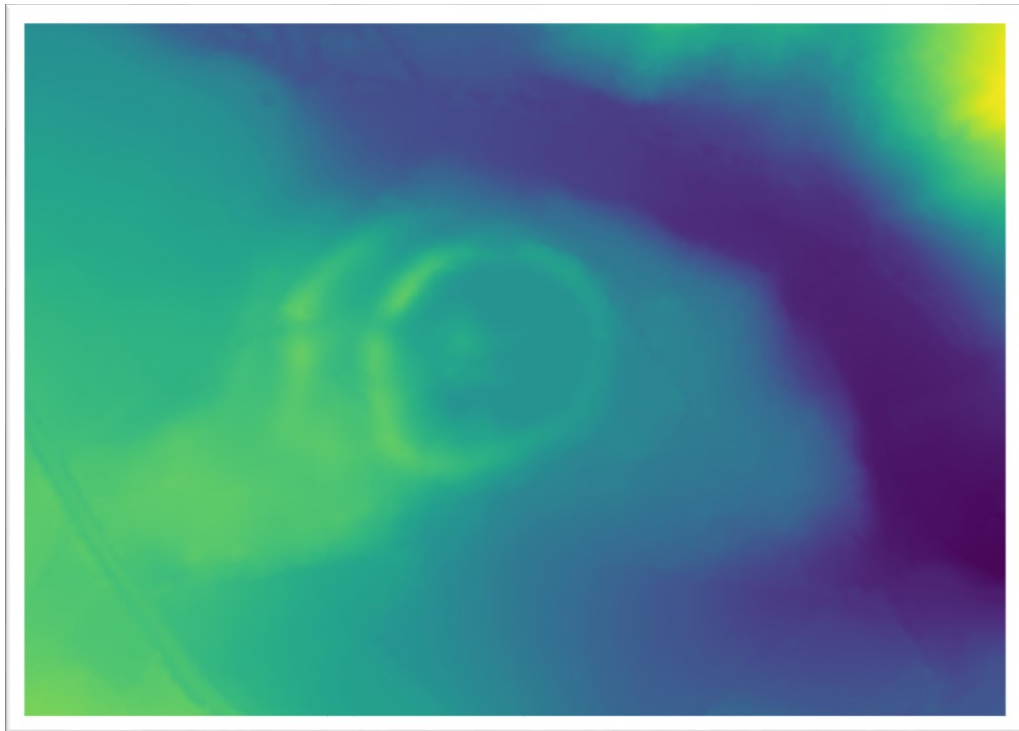


Figura 10.77. Detalle del MDE generado con el software MEL con resolución de malla de 25 cm y pseudocolor monobanda. EPSG: 25829.
Fuente: elaboración propia, a partir de datos del IGN

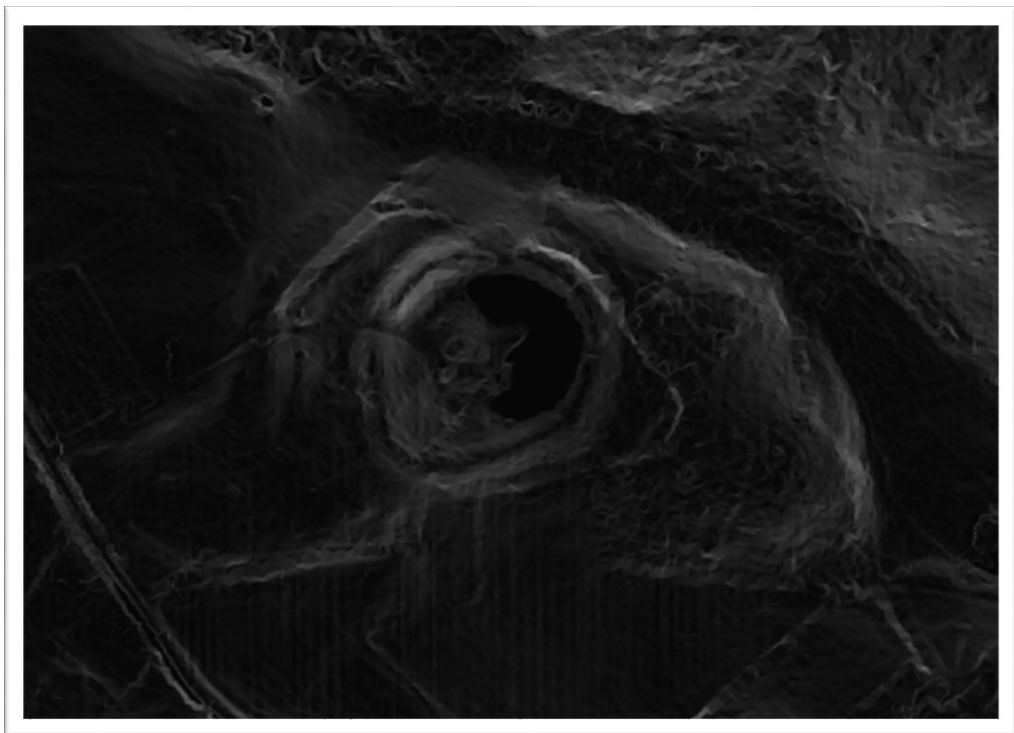


Figura 10.78. Detalle del mapa de pendientes generado con el software MEL con resolución de malla de 25 cm. EPSG: 25829
Fuente: elaboración propia, a partir de datos del IGN

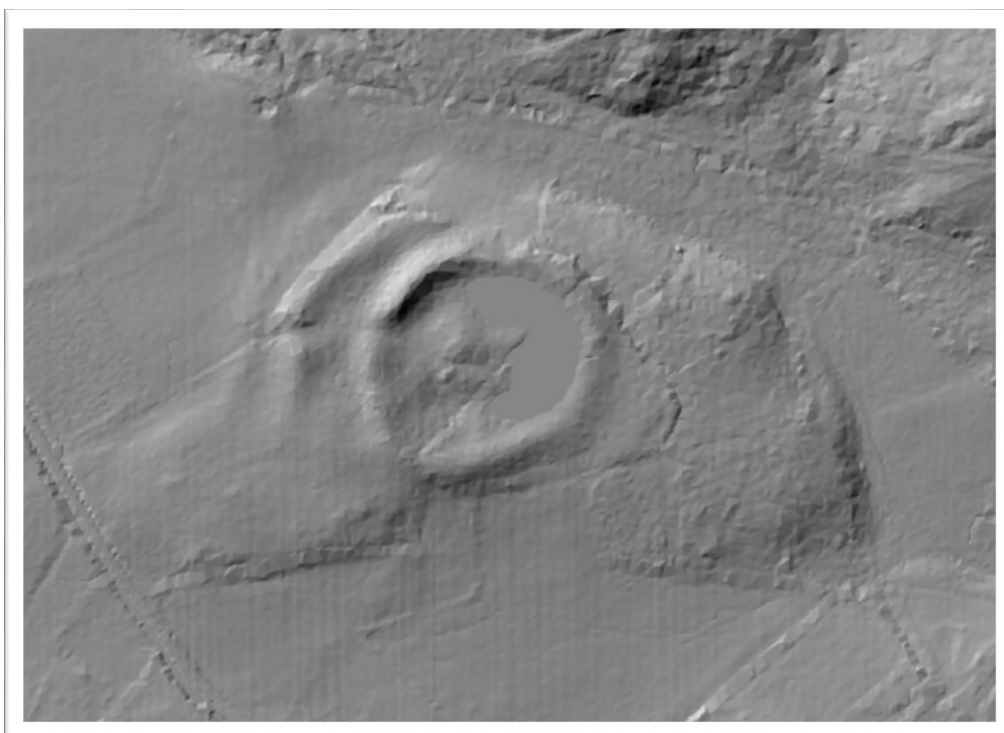


Figura 10.79. Detalle del hillshade generado con el software MEL con resolución de malla de 25 cm. EPSG: 25829

Fuente: elaboración propia, a partir de datos del IGN

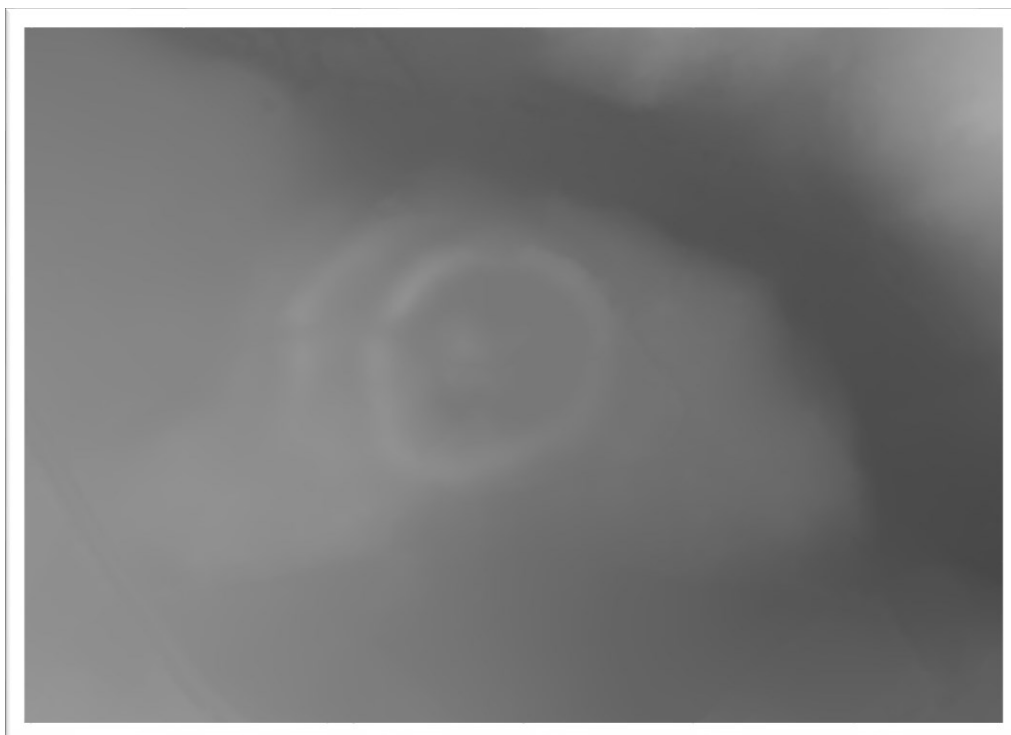


Figura 10.80. Detalle del MDE generado con el software MEL con resolución de malla de 25 cm. EPSG: 25829

Fuente: elaboración propia a partir de datos del IGN

El posible castro del Lugar de Porto Outeiro no parece estar catalogado por la Delegación de Cultura de esa Comunidad Autónoma.

En este sentido, también hay que tener en cuenta, por su importancia, la toponimia, ya que como se ha indicado en el caso del castro de Viladonga, puede dar una idea o un indicio del origen del nombre. En el caso de Lugar de Porto Outeiro, fijándose en el Mapa Topográfico Nacional 1:25.000, el posible castro estaría situado junto a un arroyo que recibe el nombre de “Rego dos Castros”, lo que podría estar indicando que junto a su cauce se levantan dos construcciones de ese tipo.

Por lo tanto, las imágenes obtenidas a partir de los datos filtrados de LiDAR permiten visualizar la estructura de un más que posible castro, con la particularidad de que el mismo no es visible en su estado natural; es decir, que debido a que la vegetación ha cubierto la práctica totalidad de la construcción esta no es apreciable en ortoimágenes aéreas como la que se muestra en la figura 10.72, correspondiente al PNOA. A pesar de lo evidente de la estructura descubierta en Lugar de Porto Outeiro durante la investigación, se carece de argumentos suficientes para confirmar que se trata realmente de un castro, a menos que se haga un estudio especializado del lugar por arqueólogos.

Al igual que ocurre con este posible castro, teniendo en cuenta las características del paisaje de la zona donde se desarrolló la cultura castreña, es presumible que muchos de los castros existentes, tan dispersos y abundantes, se encuentren totalmente ocultos y esperando ser rescatados, ofreciendo así la tecnología LiDAR una posibilidad inmejorable de facilitar esa labor. Todo ello, incluso en aquellos casos en los que el estado de la construcción no sea óptimo, pues existe un factor también importante a favor, que es el hecho de que estos castros siempre tienen unas características similares que los definen, como son su morfología y su ubicación, haciéndolos fácilmente identificables.

11. Conclusiones

Las conclusiones expuestas en este capítulo han sido posibles, gracias no sólo a las inteligencias artificiales creadas con éxito, sino también al estudio de los errores surgidos durante el proceso de entrenamiento de muchas otras que se generaron durante la investigación y al análisis de la evolución de estas.

Una de las principales conclusiones extraídas es que la aplicación de la inteligencia artificial en el campo de la cartografía digital mejora mucho los resultados obtenidos hasta ahora con algoritmos programados por humanos. Sin embargo, el entrenamiento de estas redes suele requerir un gran esfuerzo en el proceso de selección de datos y entrenamiento de estas, ya que no existían redes neuronales entrenadas específicamente para el tipo de trabajo cartográfico para el que han sido entrenadas en esta tesis. Es decir, una vez entrenado el modelo, su aplicación resulta mucho más eficiente en términos de computación y, por lo tanto, de tiempo, pero el entrenamiento resulta extremadamente costoso, razón por la cual vale la pena en el caso del tratamiento masivo de datos, pero para casos concretos habría que valorar individualmente las ventajas en los resultados frente al mayor coste computacional del entrenamiento con respecto a la aplicación de algoritmos tradicionales. Por otra parte, existen determinadas tareas que son difíciles de definir formalmente, y por lo tanto difíciles de programar, en estos casos resulta más rápido y sencillo entrenar una red neuronal con ejemplos de la tarea que se le quiere asignar que desarrollar un algoritmo que realice esa labor.

Los resultados obtenidos por modelos de Deep Learning existentes para algunas de las tareas expuestas (ya que algunos de los modelos realizan tareas que no realizaba ninguna inteligencia artificial preexistente), pueden mejorarse mucho en cuanto a su aplicación cartográfica si se entrenan modelos con datos procedentes de ortoimágenes aéreas o satelitales, ya que los modelos preexistentes no han sido entrenados con este tipo de imágenes, es decir, la

aplicabilidad de las redes neuronales en la cartografía depende en gran medida del dominio de los datos con los que fueron entrenadas. Sin embargo, pueden adaptarse diferentes dominios de las imágenes mediante el uso de redes neuronales entrenadas para tal fin, como se muestra en la siguiente imagen.



Figura 11.1. Ejemplo de adaptación de una ortoimagen del OLISTAT al PNOA
Fuente: elaboración propia, a partir de datos del IGN

El uso de redes generativas antagónicas condicionales correctamente entrenadas permite corregir errores en los datos de forma automática debido a la recreación de los datos que realizan estas redes, esta conclusión está ligada con el hecho de poder mejorar la calidad de los datos de forma automatizada cuando la mayoría de los datos presentes en el dataset de entrenamiento son correctos, lo cual tiene un impacto directo en la aplicabilidad de las redes neuronales a todo tipo de colecciones de datos geográficos de forma automatizada.

La corrección de errores en los datos resulta más efectiva cuanto más simplificada esté la imagen de entrada que se le aporta a las redes para la generación de nuevas imágenes; sin embargo, existen límites a esta simplificación, los cuales, si se sobrepasan, provocan que las imágenes

generadas, aunque realistas, no se correspondan con los elementos reales que se pretende recrear.

Por otra parte, la calidad de los resultados es mejor cuanto mayor sea la muestra de imágenes entrenada y cuantas más iteraciones se hayan realizado durante el entrenamiento, sin llegar al sobreentrenamiento del modelo. Actualmente, la única forma de establecer las iteraciones idóneas para el entrenamiento de un modelo es la revisión de los resultados y las estadísticas de entrenamiento a medida que este entrenamiento avanza.

La generación de nuevas imágenes de entrenamiento a partir de distintas combinaciones de luminosidad y contraste aplicadas a la imagen original no sólo permite aumentar los datos de entrenamiento, sino que aporta una gran versatilidad a las redes entrenadas con estos datos, lo cual es especialmente importante en el trabajo con ortoimágenes antiguas, ya que éstas presentan una gran variación de estos factores. Esto supone una innovación, no sólo en el campo de las Redes Neuronales aplicadas a la cartografía, sino en todo lo que a Redes Convolucionales se refiere.

Se ha podido comprobar que las redes entrenadas con ortoimágenes con una escala mayor devuelven mejores resultados, ya que con una resolución mayor, los elementos individuales como árboles, edificios, vehículos, etc., son más fácilmente identificables; sin embargo, tienen el inconveniente de tener que procesar un gran número de imágenes para tratar un área reducida, por lo que es necesario encontrar el equilibrio entre estos dos factores, en función del área que se vaya a tratar.

Uno de los hallazgos más interesantes es que las Inteligencias Artificiales son capaces, no sólo de aplicar el conocimiento que se les ha enseñado, sino de generar nuevo conocimiento, produciéndose una autoemergencia de comportamientos que en muchos casos resultan útiles como, por ejemplo, la

generación espontánea de nuevas categorías en los mapas de usos de superficies o la capacidad para eliminar elementos extraños en las ortoimágenes, aunque no hubiesen sido entrenadas expresamente para ello.

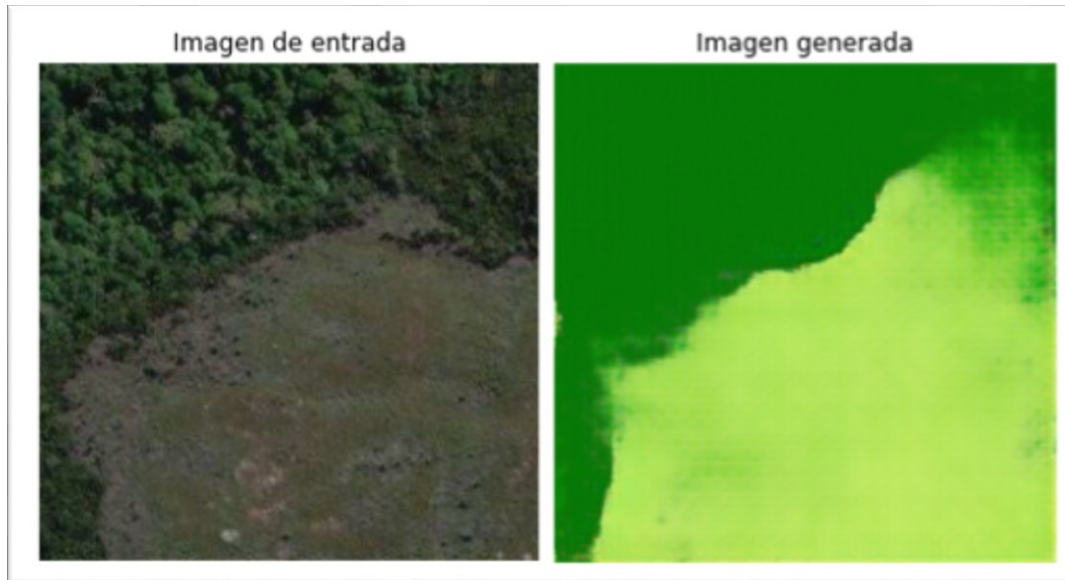


Figura 11.2. Ejemplo de comportamiento autoemergente de la Inteligencia Artificial DeepMapsUsosNatura, la cual ha creado nuevas categorías híbridas (de bosque, pastizal y brezal), representadas en forma de gradiente atendiendo a la densidad relativa de los distintos elementos presentes (árboles, arbustos y pastos). (Escala 1:1.300)
Fuente: Elaboración propia, a partir de los datos de la capa Esri Aéreo

El conocimiento generado por los humanos en un área concreta, como es el caso de la limitación del Copernicus Land Monitoring Service y del CORINE, los cuales están limitados a Europa, puede extenderse a cualquier lugar del planeta mediante el uso de la Inteligencia Artificial; un ejemplo de esto lo tenemos en la imagen superior, en la cual se muestra cómo, a partir de una ortoimagen de la capa Esri Aéreo del South Western National Park de Tasmania, se ha generado un mapa de usos del suelo con las características de los mapas de usos de la Red Natura2000, mediante la utilización de la Inteligencia Artificial DeepMapsUsosNatura. El área representada en el ejemplo anterior se corresponde con las siguientes coordenadas: 16260543.435726378, 16260938.257560115, -5347689.786458732, -5347288.148958321 [EPSG: 102100]

Sin embargo, dependiendo del entrenamiento que reciban las inteligencias artificiales, pueden llegar a un punto de abstracción que deja de resultar útil a nuestros propósitos, convirtiendo las capacidades que han emergido de forma espontánea en un inconveniente, en lugar de una virtud.

Otra conclusión por destacar que se ha podido obtener de los distintos entrenamientos es que, si bien los modelos van mejorando a medida que son entrenados, si el entrenamiento se realiza con una misma muestra de imágenes llega un punto en el que, o bien los resultados empiezan a empeorar o bien la red se vuelve demasiado específica, teniendo dificultades al aplicarla en imágenes distintas a las de la muestra de entrenamiento; cuanto más se sobreentrene una red más específica se vuelve. Esto también depende, en gran medida, del número de imágenes utilizadas durante el entrenamiento y la variedad de estas.

Existe un punto óptimo en el entrenamiento de los modelos, que no siempre es el que aporta mejores resultados en el set de datos utilizado para las pruebas, ya que un exceso de entrenamiento acaba especializando demasiado el modelo, este exceso de especialización puede solventarse, en gran medida, incluyendo ruido aleatorio en los datos de entrenamiento.

En esta tesis también se ha comprobado que es posible generar mapas de ocupación del suelo utilizando Redes Neuronales entrenadas con los datos adecuados.

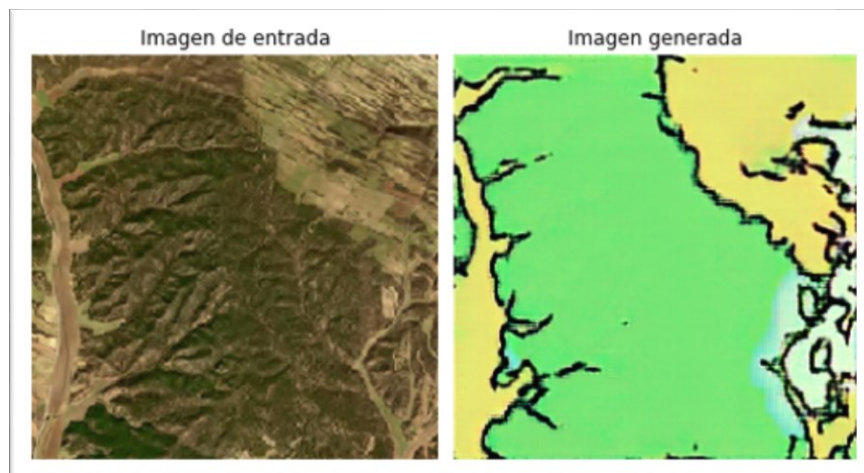


Figura 11.3. Mapa de usos generado por la inteligencia artificial DeepMapsUsosSIOSE
Fuente: elaboración propia, a partir de datos del IGN

Pero este tipo de redes son más susceptibles al sobreentrenamiento, debido a que los datos de salida de la red son más simples y, por lo tanto, existe menos variedad posible. Cuando estas redes son entrenadas durante demasiadas iteraciones acaban dando lugar a patrones erróneos que se repiten, tal y como se muestra en los siguientes ejemplos, es decir un modelo entrenado para generar imágenes de sólo dos colores será sobreentrenado y dará lugar a errores con un número menor de iteraciones que otro entrenado para generar imágenes con más colores.





Figura 11.4. Ejemplo de errores producidos por un modelo sobreentrenado
Fuente: elaboración propia, a partir de datos del IGN

El pretratamiento de las imágenes con las que son entrenadas las redes neuronales puede mejorar su rendimiento y aumentar su tolerancia a errores.

La selección adecuada de imágenes para el entrenamiento de redes neuronales tiene un efecto directo en el rendimiento de estas; para obtener esta conclusión, ha sido fundamental la revisión de resultados en redes que han sido entrenadas con imágenes sin seleccionar y que han dado lugar a algunos errores en los resultados.

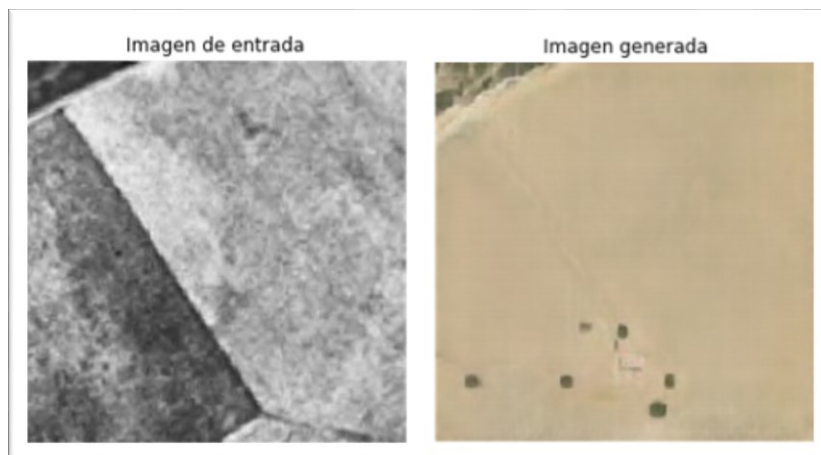


Figura 11.5. Ejemplo de resultado obtenido con una de las redes entrenadas sin una selección correcta de las fuentes

Fuente: elaboración propia, a partir de datos del IGN

La inclusión de ruido en las imágenes de entrenamiento de las redes neuronales destinadas a mejorar el enfoque de las ortoimágenes aéreas proporciona mayor versatilidad y permite un entrenamiento más rápido, ya que la variedad de muestras necesarias es menor.

La simplificación de los datos en el proceso de entrenamiento, mediante el preprocesado de los mismos, para generar imágenes de los bordes permite la adaptación de dominios, con la cual pueden aplicarse modelos entrenados con datos pertenecientes a diferentes dominios y, por lo tanto, extender la aplicabilidad de las redes neuronales cartográficas.

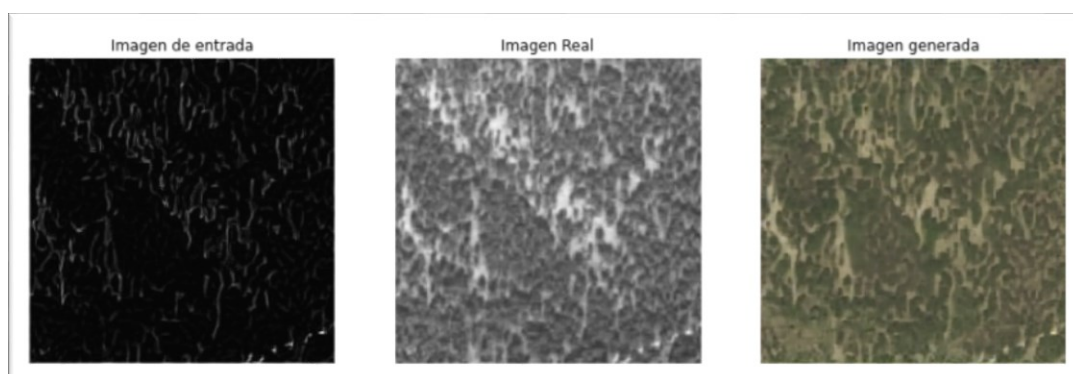


Figura 11.6. Adaptación de dominio de ortoimagen del OLISTAT a PNOA realizada por el modelo DeepMapsBordes

Fuente: elaboración propia, a partir de datos del IGN

Algunas de las redes neuronales desarrolladas, además de aplicaciones puramente cartográficas, tendrán aplicaciones militares, ya que son capaces tanto de corregir como de mejorar las actuales técnicas de censura u ocultamiento en ortoimágenes aéreas y satelitales. El modelo DeepMapsInpainting, además de su utilidad para corrección de fallos, tiene potenciales aplicaciones militares; esta Inteligencia Artificial es capaz de completar las áreas carentes de datos, debidos a la aplicación de parches de censura, con datos verosímiles, aportando un aspecto mucho mejor a la imagen y dificultando enormemente la labor de identificar aquellas áreas sensibles que habían sido originalmente censuradas. El capítulo 5 muestra, además de las correcciones de las censuras, algunos ejemplos de la aplicación del modelo DeepMapsOculta, también de creación propia, el cual es capaz de ocultar edificios y estructuras de creación humana como elementos naturales, lo cual tiene claras aplicaciones militares y resuelve la problemática de censurar determinadas zonas de forma mucho más verosímil y difícil de identificar que mediante las técnicas basadas en la aplicación de parches o difuminados utilizadas actualmente.

Tras la revisión de los resultados obtenidos durante las investigaciones realizadas con datos tridimensionales, se puede concluir que, tal y como se había propuesto en las hipótesis, es posible generar un modelo digital del terreno de mayor resolución al ofrecido por el IGN, a partir de los datos obtenidos por los vuelos LiDAR del Plan Nacional de Ortofotografía Aérea.

Esto queda demostrado con la elaboración de los Modelos Digitales del Terreno obtenidos con el software MEL, desarrollado para esta investigación, con el cual se han generado MDE con una resolución de malla de 0,25 metros, frente a la resolución de malla de 5 metros ofrecida por el MDT05 suministrado por el IGN o incluso la más reciente de 2 metros (MDT02), tal y como se aprecia en las imágenes que hay a continuación. Estas imágenes muestran un campo de placas solares donde se comprueba cómo la resolución ofrecida por el MDT del IGN es insuficiente para identificar tridimensionalmente determinados

elementos, los cuales, sin embargo, se muestran con claridad mediante el tratamiento de datos de la nube de puntos realizado con el software MEL.



Figura 11.7. Imagen del PNOA.
Fuente: elaboración propia, a partir de datos del IGN



Figura 11.8. Detalle del MDT05 suministrado por el IGN. EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectoado)
Fuente: elaboración propia, a partir de datos del IGN

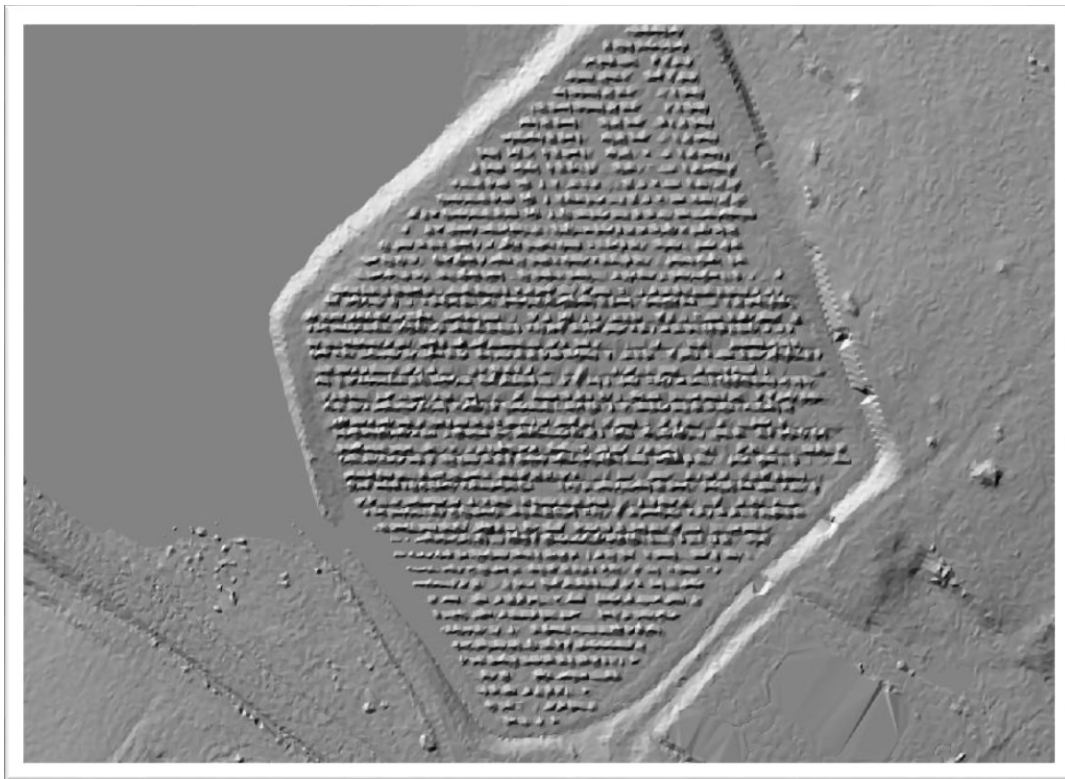


Figura 11.9. Detalle del hillshade de resolución de malla de 25 cm obtenida con el software MEL sobre cartografía del IGN. EPSG: 25830 (ETRS89 / UTM zona 30N - Proyectado).
Fuente: elaboración propia, a partir de datos del IGN

Asimismo, ha quedado demostrado que se puede desarrollar un software libre, que facilite el análisis del relieve, mediante la automatización de la generación de archivos de forma masiva, derivados de los ficheros LiDAR ofrecidos por el IGN, con diferentes parámetros de altitud, azimut y factor z, etc. Esto se ha plasmado en creación de varias herramientas de software libre desarrolladas y utilizadas en esta tesis, principalmente el software online para análisis del relieve (<http://cartotecnia.com/teledeteccion/>) y el software MEL centrado en el análisis de los datos contenidos en los ficheros LiDAR y la generación de varios tipos de archivos geográficos derivados de los mismos.

Durante el desarrollo de las investigaciones que se han realizado para la elaboración de este trabajo, se ha podido comprobar que los diversos softwares elaborados son adecuados tanto para el estudio del relieve, la geomorfología regional y la identificación de estructuras de relevancia histórico-arqueológica en

España, siendo crucial en algunos casos la gran resolución obtenida, que permite mostrar elementos que quedarían ocultos con las resoluciones de malla ofrecidas por el IGN. Además, resulta fundamental la capacidad de filtrado de datos proporcionada por el software MEL para poder identificar estructuras que se encuentran cubiertas por vegetación, lo cual impide su apreciación mediante técnicas más tradicionales como la fotografía aérea.

Un buen ejemplo de esto es la diferencia que se aprecia entre las imágenes obtenidas a partir del MDT05 del IGN y las generadas por el software MEL, las cuales permiten detectar estructuras, como en el caso de estudio de un posible campamento romano en Hiendelaencina (Guadalajara), que pasan desapercibidas con los actuales MDT utilizados o con la fotografía aérea debido a la presencia de vegetación. A continuación, se incluyen, como ejemplo de lo mencionado, varias imágenes de la misma zona obtenidas durante su análisis con la herramienta de escritorio creada. Las dos primeras se corresponden con la imagen del PNOA y el MDT05 del IGN, mientras que las siguientes han sido obtenidas con el software desarrollado en esta investigación y son las que han permitido detectar el posible campamento romano, ya que uno de los elementos que se revelan es una clara “claviculae” o puerta en clavícula.

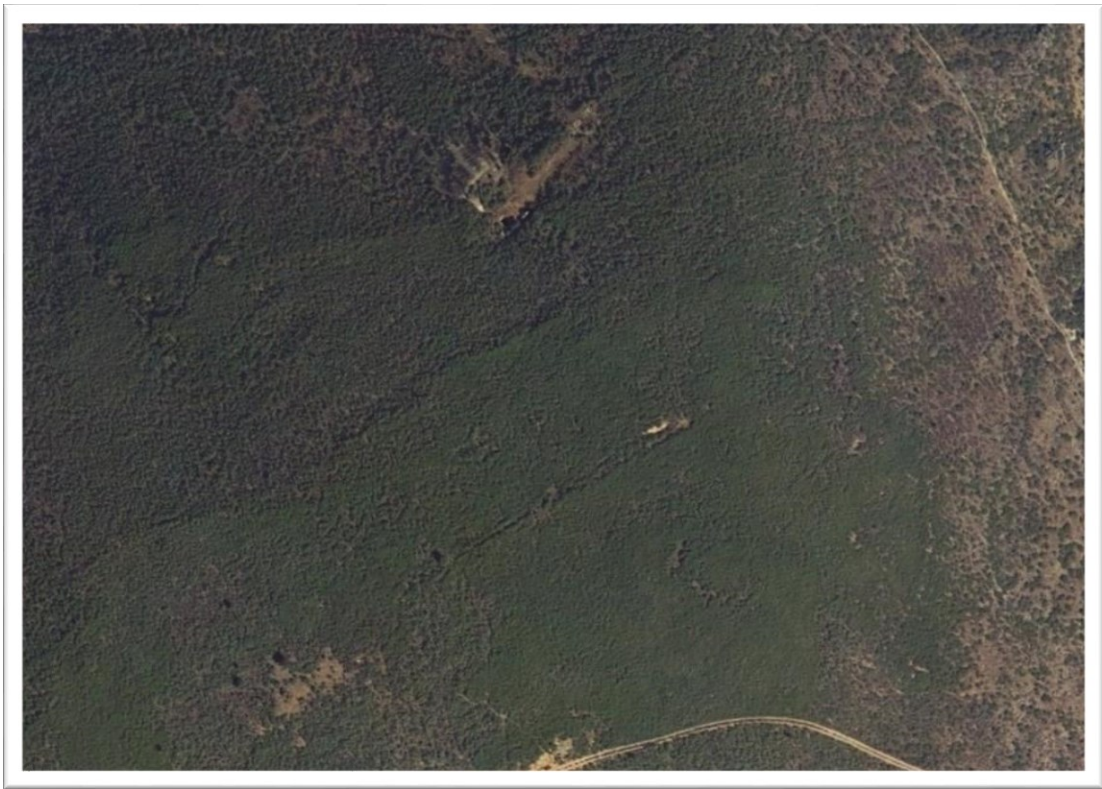


Figura 11.10. Imagen del PNOA de Máxima Actualidad, del posible campamento romano de Hiendelaencina.

Fuente: elaboración propia, a partir de datos del IGN

Al comparar las dos imágenes expuestas a continuación, resulta evidente la mejora producida en los resultados mediante la aplicación del software desarrollado, el cual permite apreciar muchos detalles que pasarían inadvertidos utilizando los MDT suministrados por el IGN.

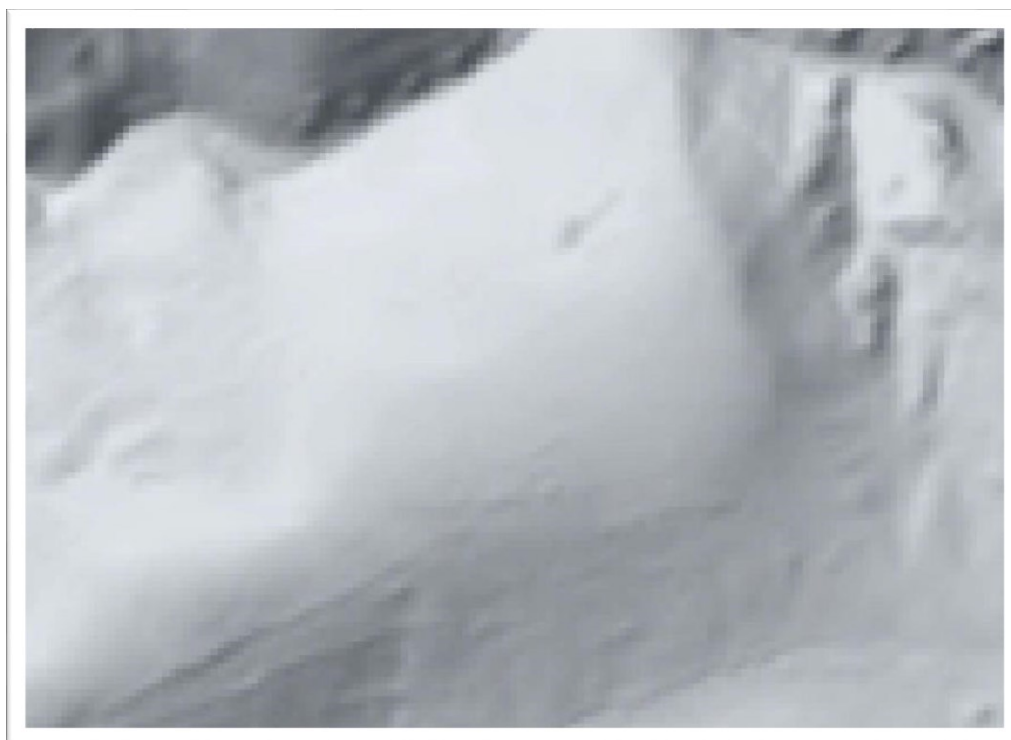


Figura 11.11. MDT05 del IGN, del posible campamento romano de Hiendelaencina.
Fuente: elaboración propia, a partir de datos del IGN

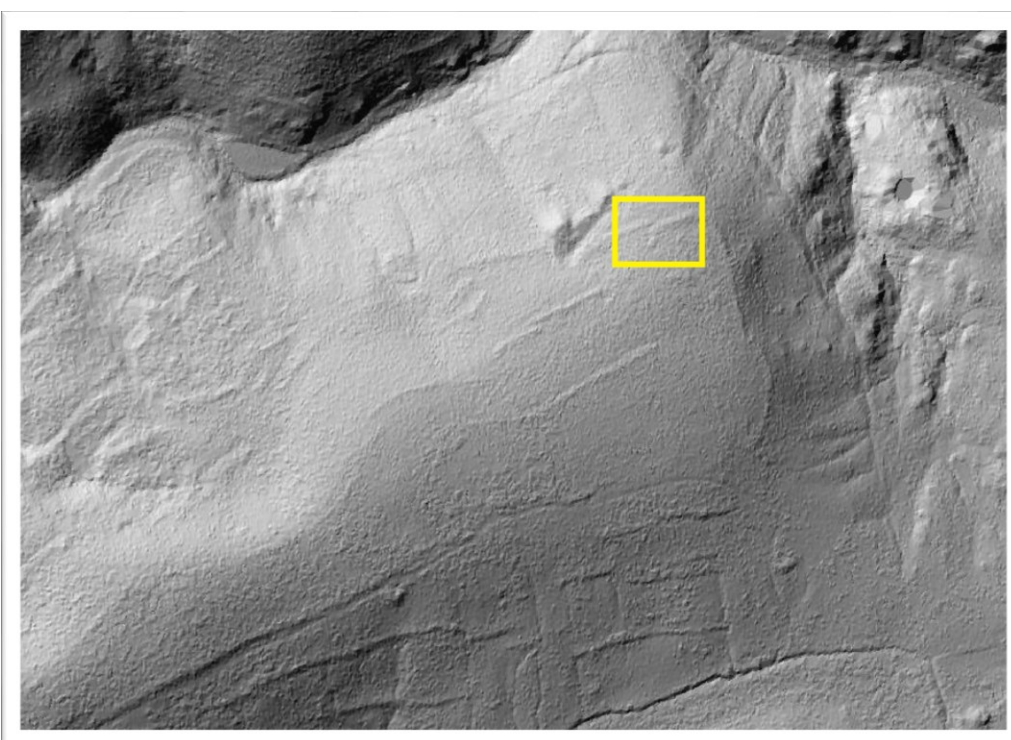


Figura 11.12. Hillshade generado con el software “Morillas Easy LiDAR”, del posible campamento romano de Hiendelaencina. EPSG:25830 (ETRS89 / UTM zona 30N - Proyectado), donde puede apreciarse la clavícula
Fuente: elaboración propia, a partir de datos del IGN

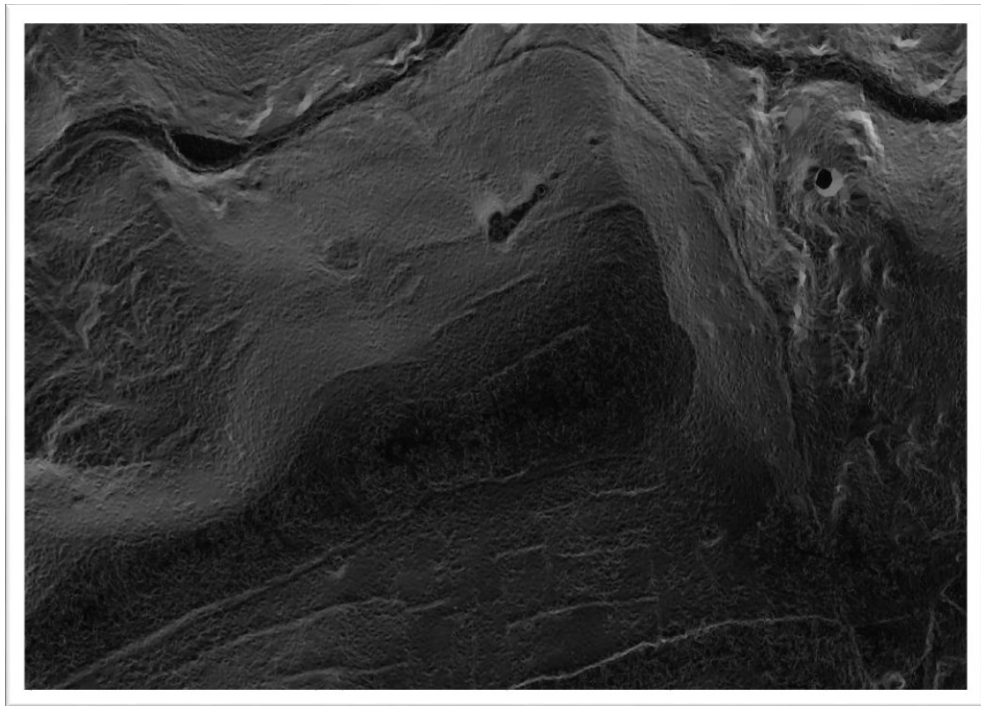


Figura 11.13. Mapa de pendientes generado con el software “Morillas Easy LiDAR”, del posible campamento romano de Hiendelaencina. EPSG:25830 (ETRS89 / UTM zona 30N - Proyectado), donde puede apreciarse la clavícula
Fuente: elaboración propia, a partir de datos del IGN

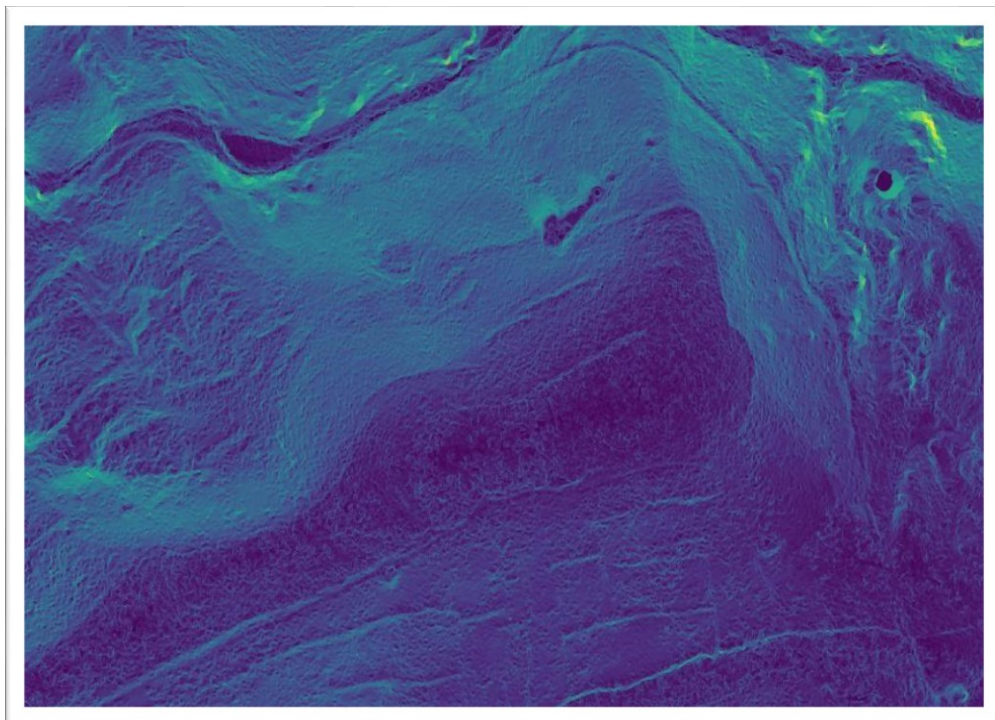


Figura 11.14. Mapa de pendientes generado con el software “Morillas Easy LiDAR”, con pseudocolor monobanda, del posible campamento romano de Hiendelaencina. EPSG:25830 (ETRS89 / UTM zona 30N - Proyectado), donde puede apreciarse la clavícula
Fuente: elaboración propia, a partir de datos del IGN

12. Anexos

12.1. Anexo I: Evolución de las inteligencias artificiales creadas

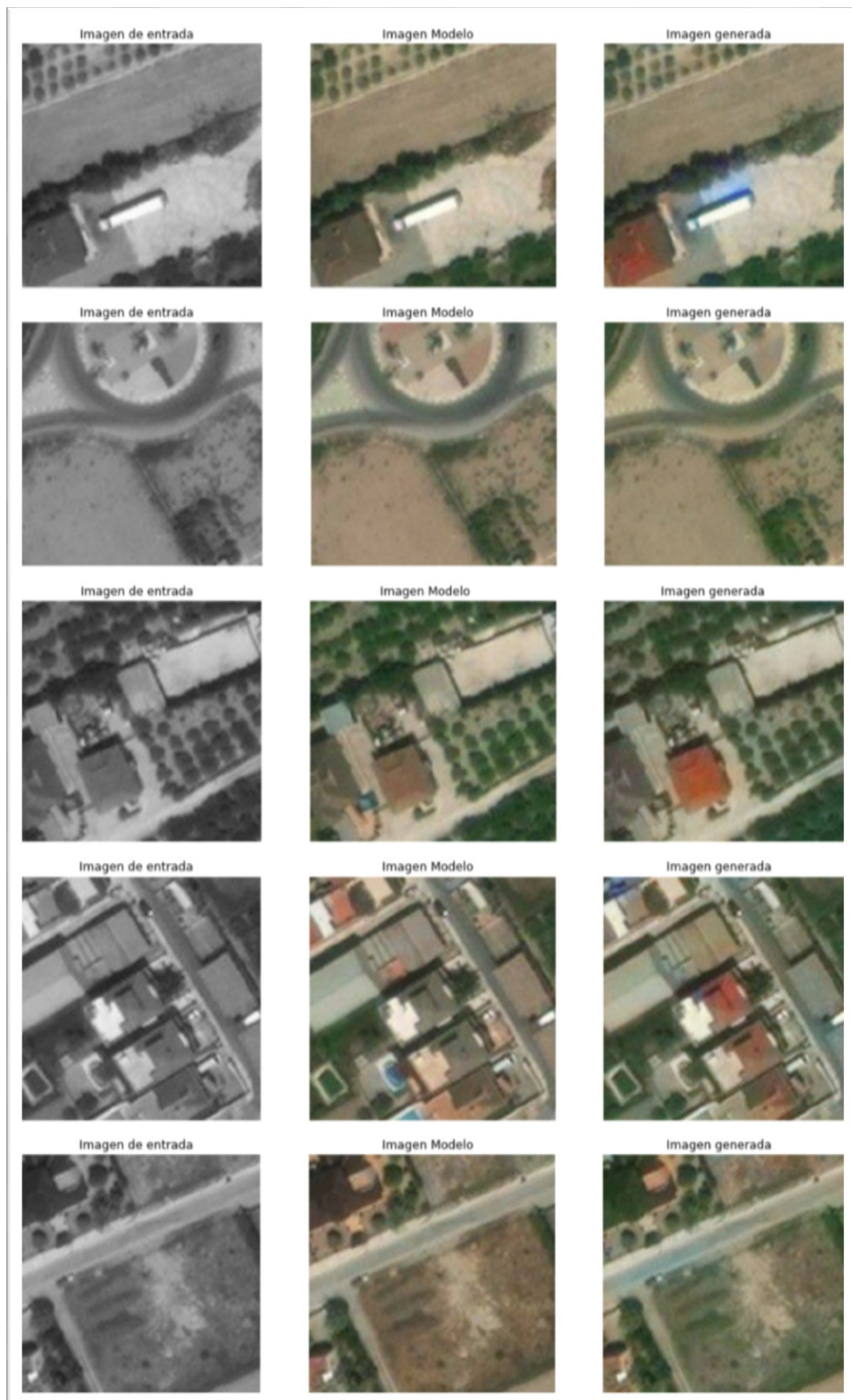
Este anexo incluye una muestra de las imágenes generadas por algunas de las inteligencias artificiales en distintas fases de su entrenamiento, todas ellas fases tempranas. El análisis de la evolución de los distintos modelos ha permitido extraer las conclusiones necesarias para poder crear las inteligencias artificiales expuestas en este documento y obtener mejores resultados que los proporcionados por cualquiera de los métodos existentes con anterioridad.

12.1.1. Ejemplo del proceso de entrenamiento del modelo DeepMapsColor19

A continuación, se muestran los progresos realizados durante el aprendizaje de la inteligencia artificial creada para dar color a teselas en blanco y negro en una escala aproximada de 1:1.000. Éste es sólo un ejemplo de cómo han ido evolucionando las inteligencias artificiales desarrolladas durante esta investigación a medida que han avanzado en su entrenamiento. Todas las inteligencias creadas partieron de cero, por lo que antes de comenzar su entrenamiento devolvían datos incoherentes (como en la primera imagen del proceso de entrenamiento mostrado tras este párrafo), sin embargo en sólo unas pocas iteraciones aprendían la tarea para la que estaban siendo entrenadas, volviéndose más expertas y aportando mejores resultados a medida que el entrenamiento avanzaba, llegando todas ellas a superar cualquiera de los sistemas o procedimientos existentes orientados a los mismos fines, sobre todo en cuanto a su aplicación cartográfica se refiere.

En los siguientes ejemplos puede apreciarse cómo la inteligencia artificial, según es entrenada durante más iteraciones va devolviendo resultados cada vez más precisos.







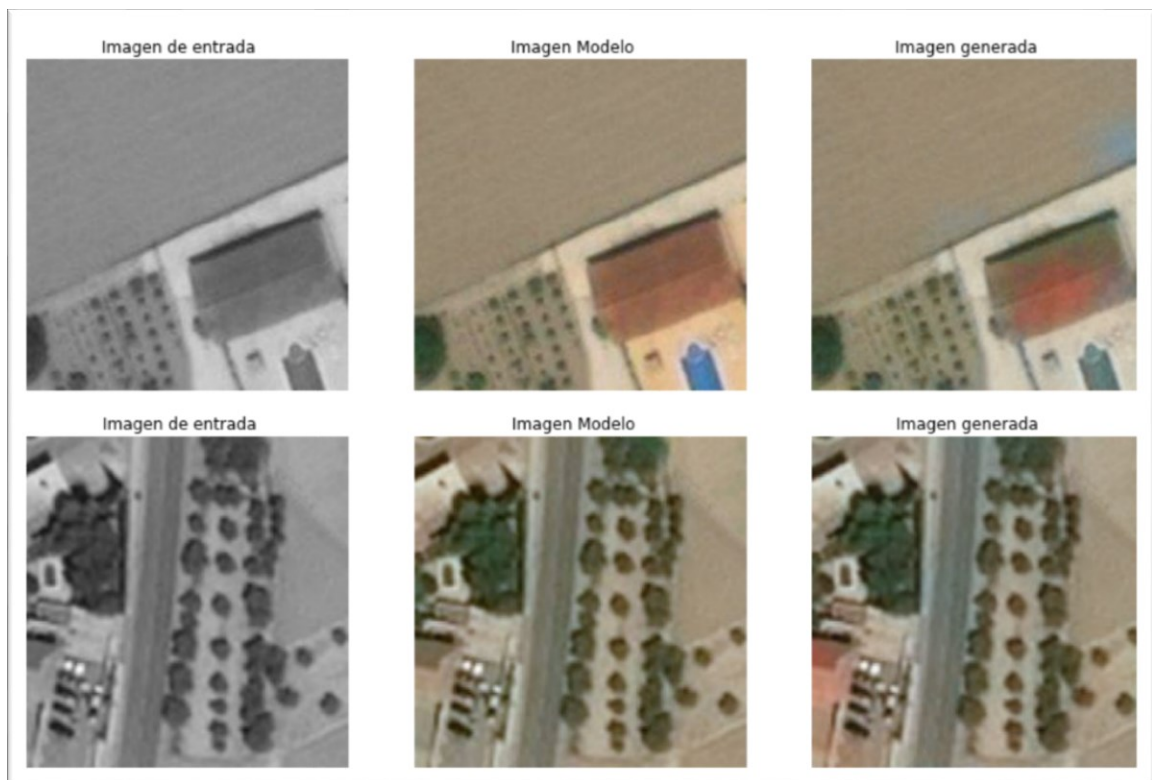


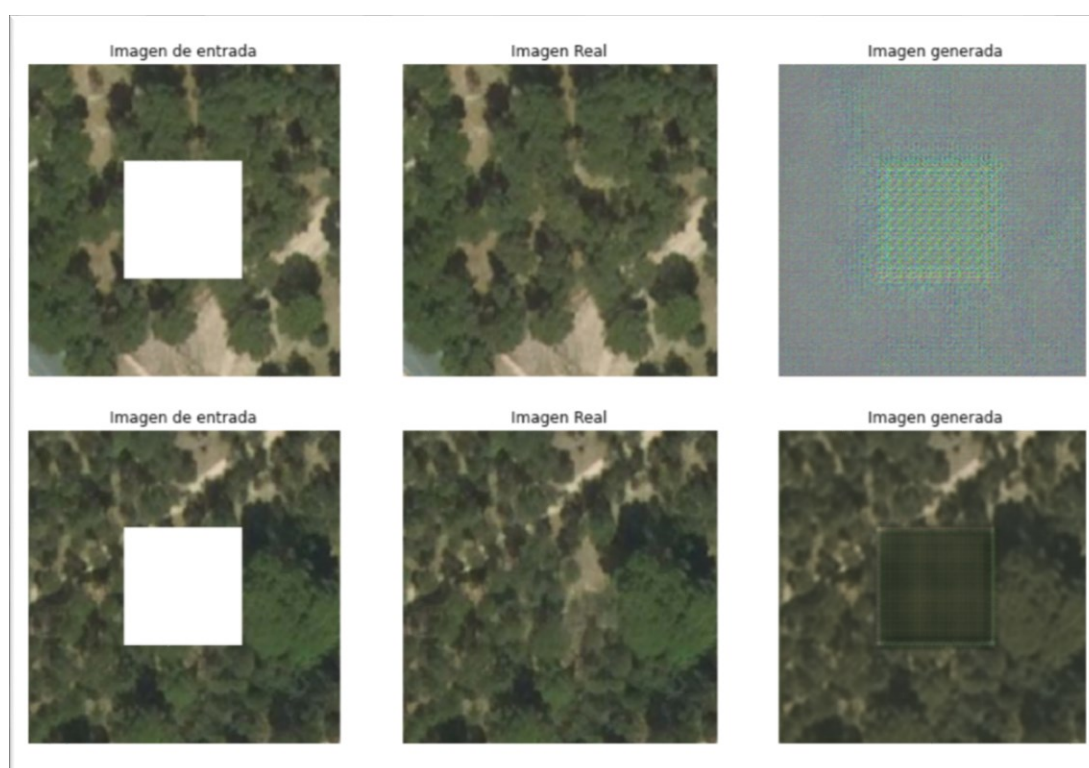
Figura 12.1. Imágenes de los resultados aportados por el modelo DeepMapsColor19 en distintas etapas de su entrenamiento

Fuente: elaboración propia, a partir de datos del IGN

Durante el entrenamiento de esta inteligencia artificial puede apreciarse cómo, al inicio de éste, las representaciones generadas ni siquiera reflejaban los elementos presentes en la imagen original. A medida que aumentaban las iteraciones la red iba aprendiendo a representar los distintos elementos y tras esto aprende a colorearlos. El aprendizaje de la coloración de los distintos elementos presentes en las imágenes no se produce todo a la vez; en primer lugar, se ha observado cómo la red comienza a distinguir la vegetación y, por lo tanto, a colorearla. Pasado un tiempo, comienza a ser capaz de colorear correctamente otros elementos como las carreteras; tras esto aprende a colorear los tejados y, por último, acaba siendo capaz de colorear elementos menos frecuentes como las piscinas o los coches. Hay que mencionar que, si bien la coloración generada es mucho más realista que la obtenida con cualquiera de los métodos existentes, como el color de los coches o los tejados, no siempre es el mismo, a pesar de que la coloración generada por nuestra inteligencia

artificial es completamente verosímil y podría engañar a cualquier humano, no tiene por qué corresponderse con el color original, ya que existen diferentes colores posibles en tejados o coches, los cuales al ser pasados a escala de grises resultan similares, por lo que el modelo desarrollado genera las coloraciones más probables en función de la forma y el tono presente en la imagen en escala de grises con la que ha sido alimentada.

12.1.2. Ejemplo del proceso de entrenamiento del modelo DeepMapsInpainting



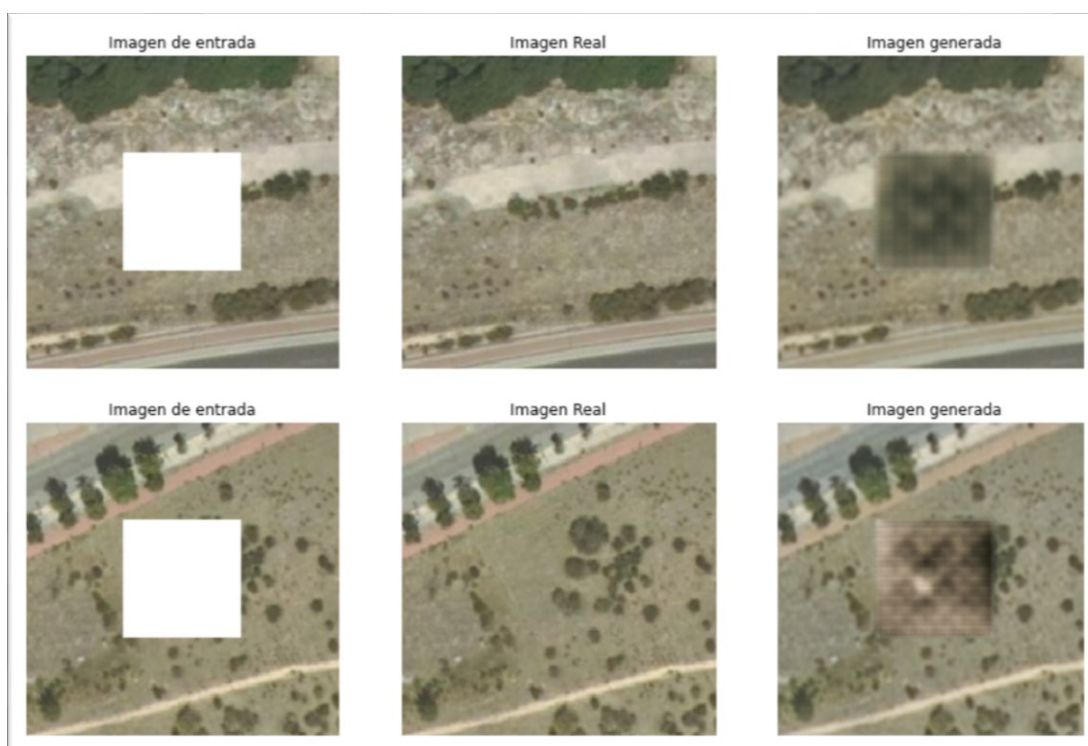
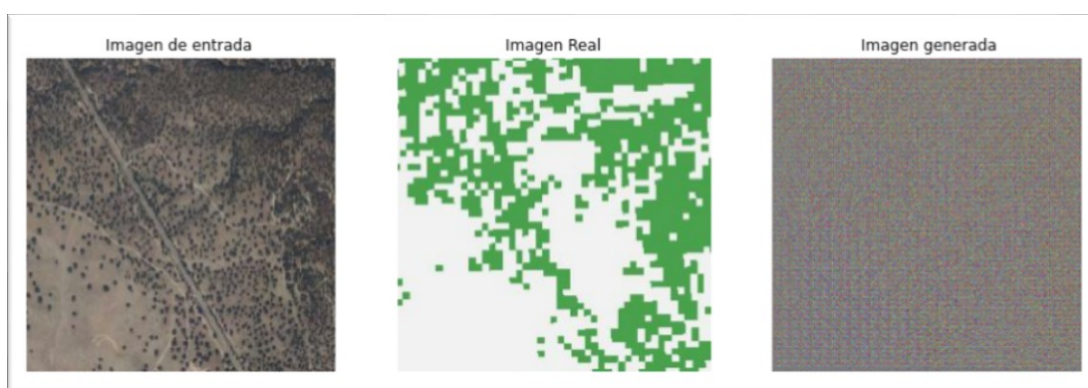


Figura 12.2. Imágenes de los resultados aportados por el modelo DeepMapsInpainting en distintas etapas de su entrenamiento
Fuente: elaboración propia, a partir de datos del IGN

12.1.3. Ejemplo del proceso de entrenamiento del modelo DeepMapsBosques



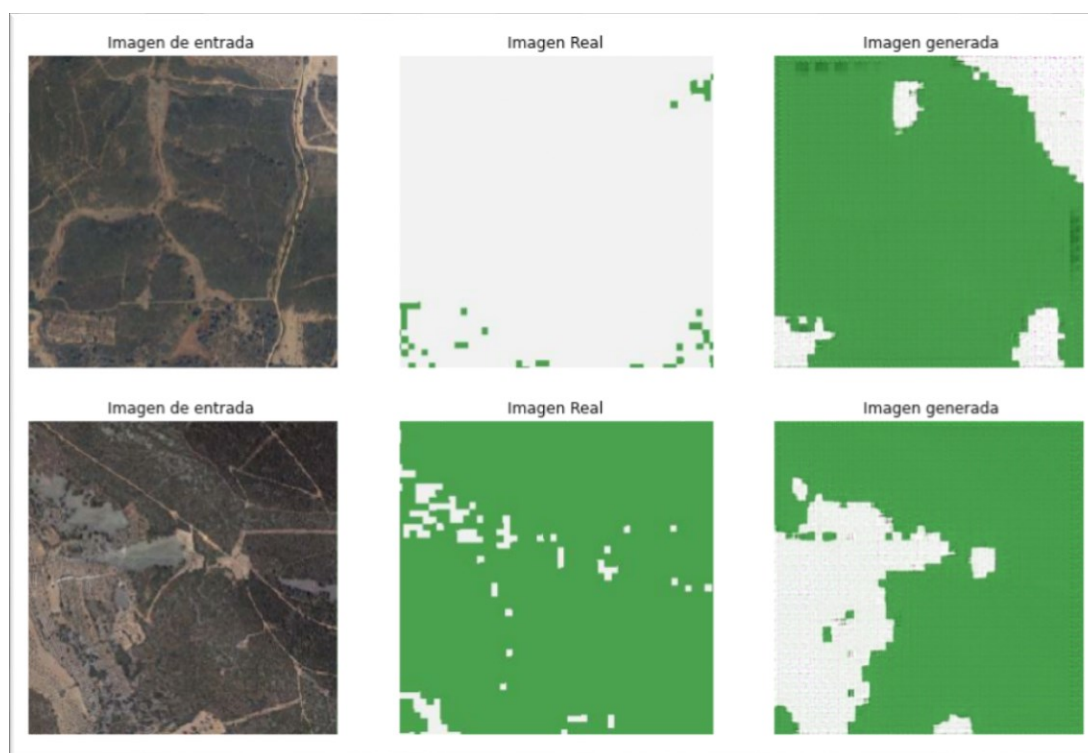


Figura 12.3. Imágenes de los resultados aportados por el modelo DeepMapsBosques en distintas etapas de su entrenamiento

Fuente: elaboración propia, a partir de datos del IGN

En este ejemplo puede apreciarse como a medida que el entrenamiento ha avanzado, la inteligencia artificial DeepMapsBosques va mejorando en su precisión a la hora de reconocer la cobertura de árboles, llegando en las últimas etapas a una precisión mayor que la presente en los datos originales con los que ha sido entrenada (Imagen Real).

12.1.4. Ejemplo del proceso de entrenamiento del modelo DeepMapsBordes17

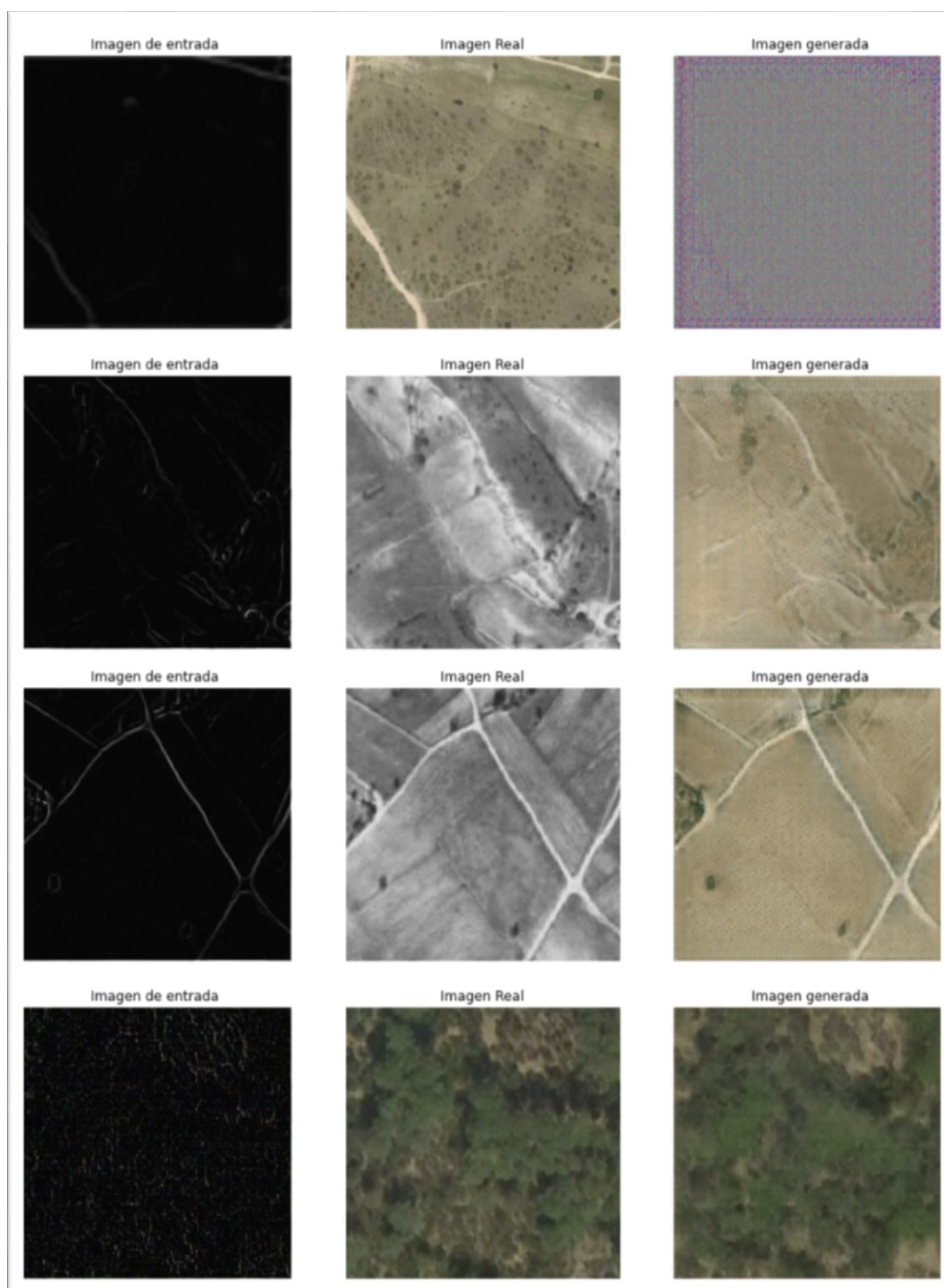
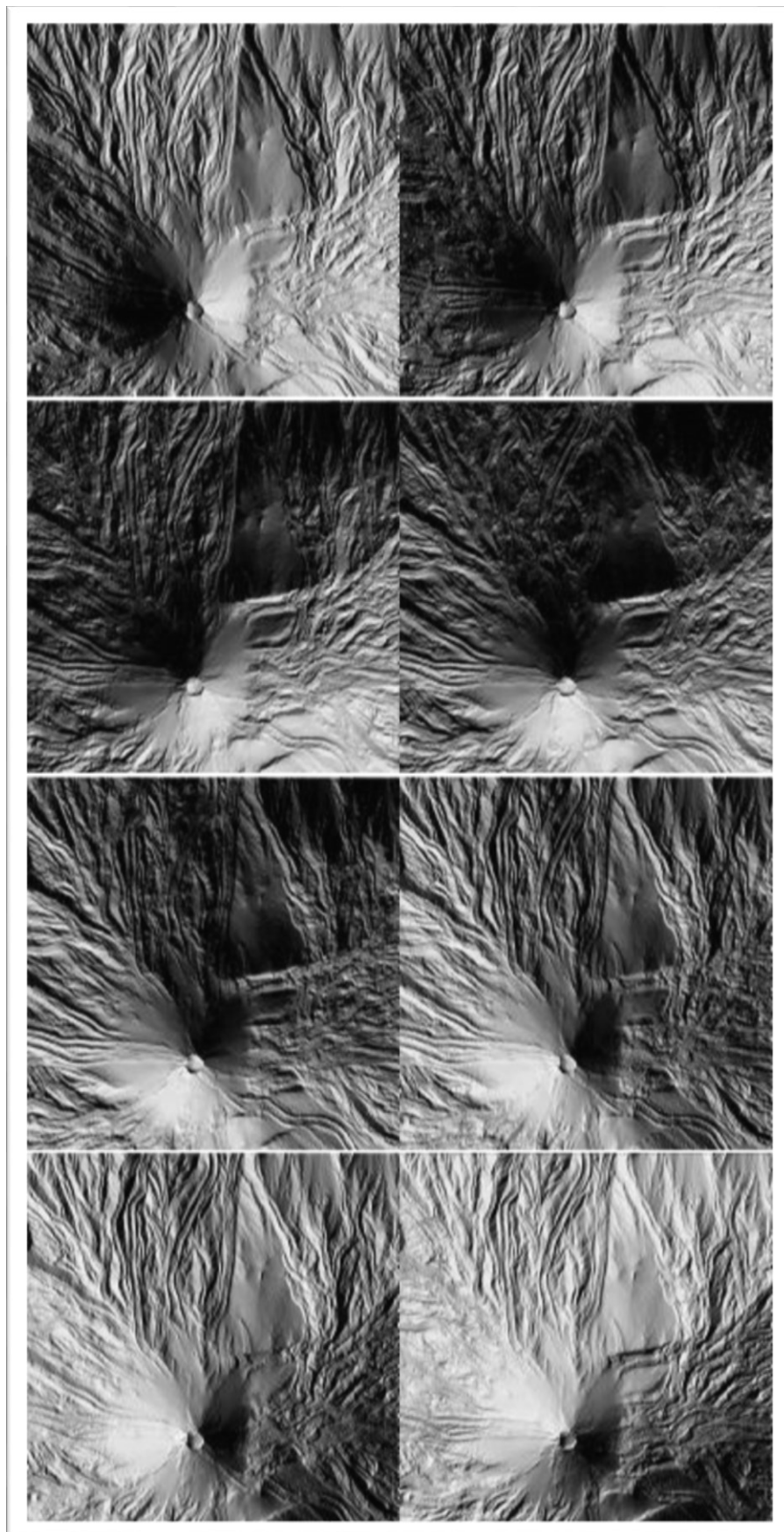


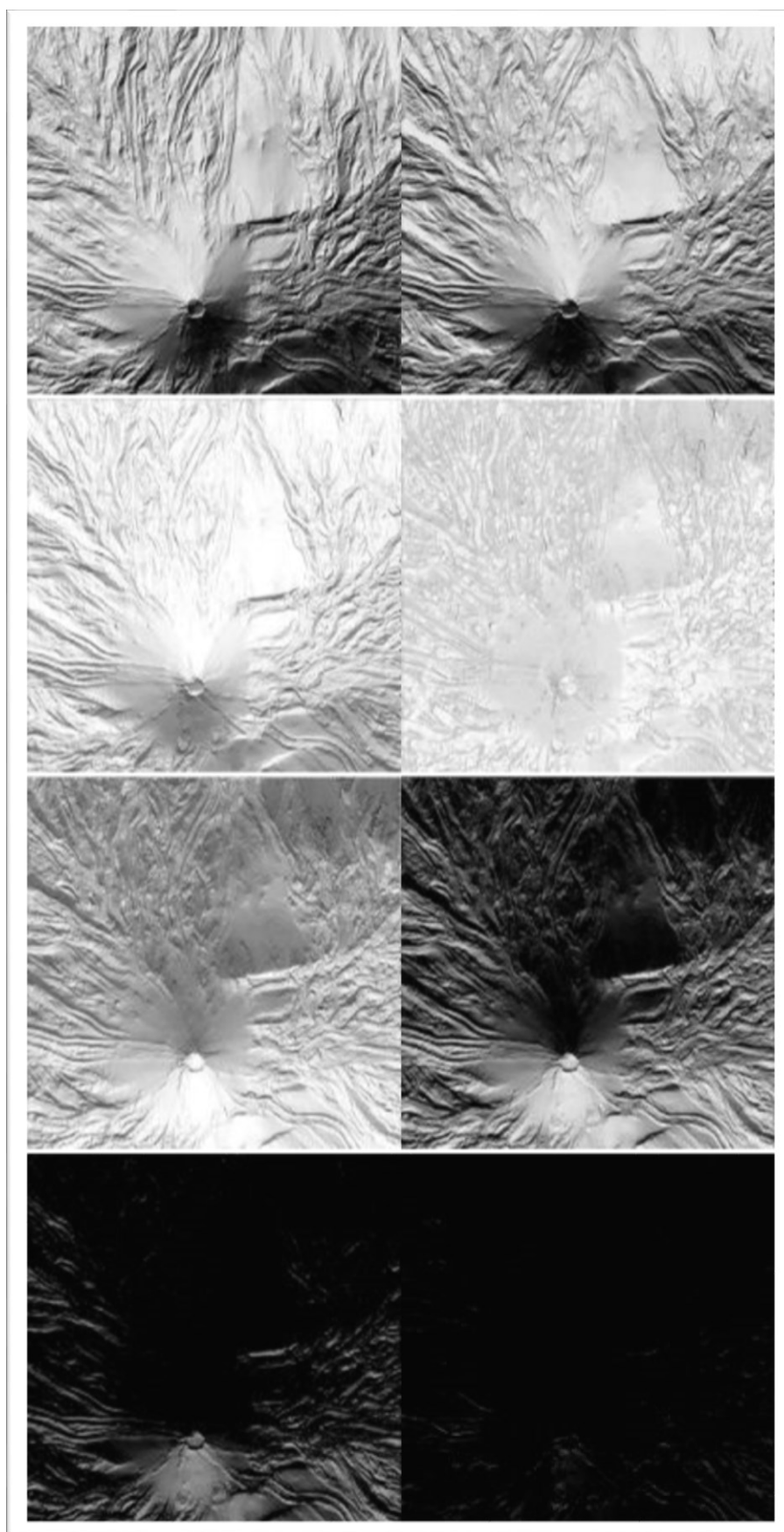
Figura 12.4. Imágenes de los resultados aportados por el modelo DeepMapsBordes17 en distintas etapas de su entrenamiento.

Fuente: elaboración propia, a partir de datos del IGN.

12.2. Anexo II: Hillshades multiparámetro y mapas hipsométricos multialtitud

En este anexo se muestra un ejemplo de las imágenes obtenidas al ejecutar las funciones de generación de hillshades multiparámetro y de generación de mapas hipsométricos multialtitud incluidas en el software MEL, debe añadirse que a día de hoy este software es el único que incluye esta función, la cual permite un gran ahorro de tiempo a la hora de seleccionar los parámetros más adecuados para la creación tanto de un hillshade de un área determinada como del mapa hipsométrico correspondiente a ese área. A continuación se expone una muestra de parte de los archivos generados en este proceso, en la generación de hillshades multiparámetro y de mapas hipsométricos multialtitud se puede seleccionar también la resolución de malla con la que se quieren producir las muestras, en este caso se ha utilizado una resolución de malla de 1m por pixel, que si bien no es la máxima resolución que permite el software generado, si es mayor a la proporcionada por el IGN y su producción resulta más rápida que otras resoluciones mayores. Las imágenes mostradas en este anexo han sido obtenidas a partir del archivo LiDAR PNOA_2009_Lote_CAN-Ten_338-3130_ORT-CLA-COL.laz, suministrado por el IGN, el cual recoge el estratovolcán del Teide situado en la isla de Tenerife (Canarias, España).





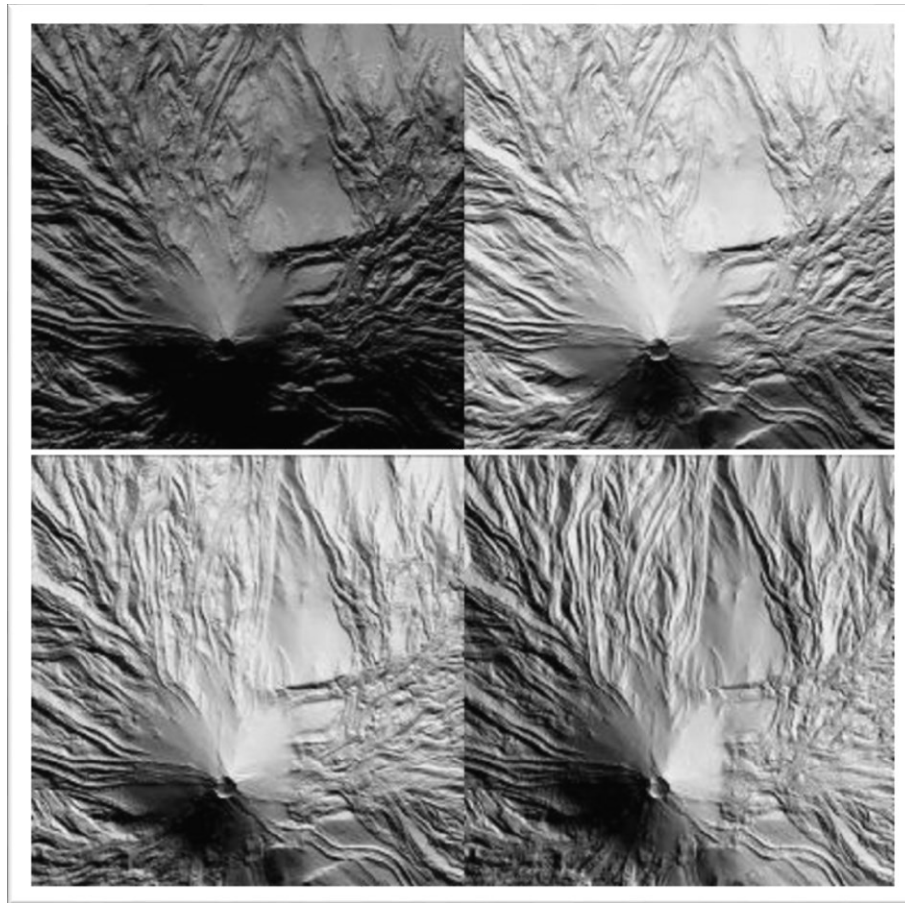
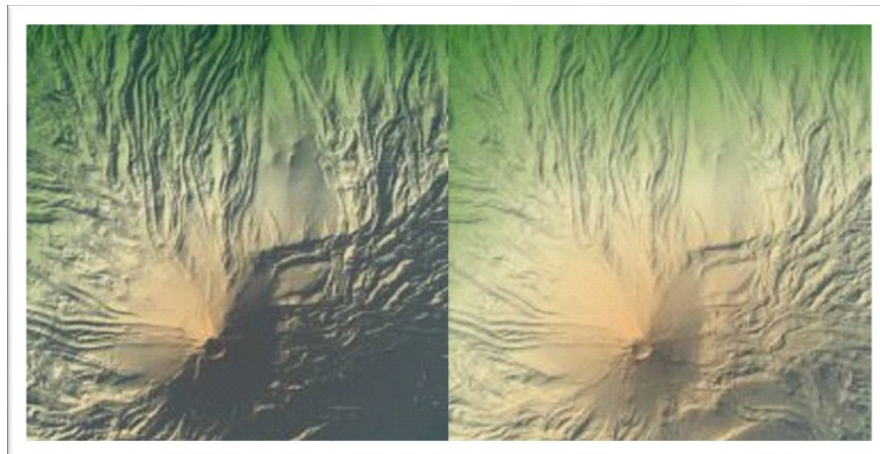
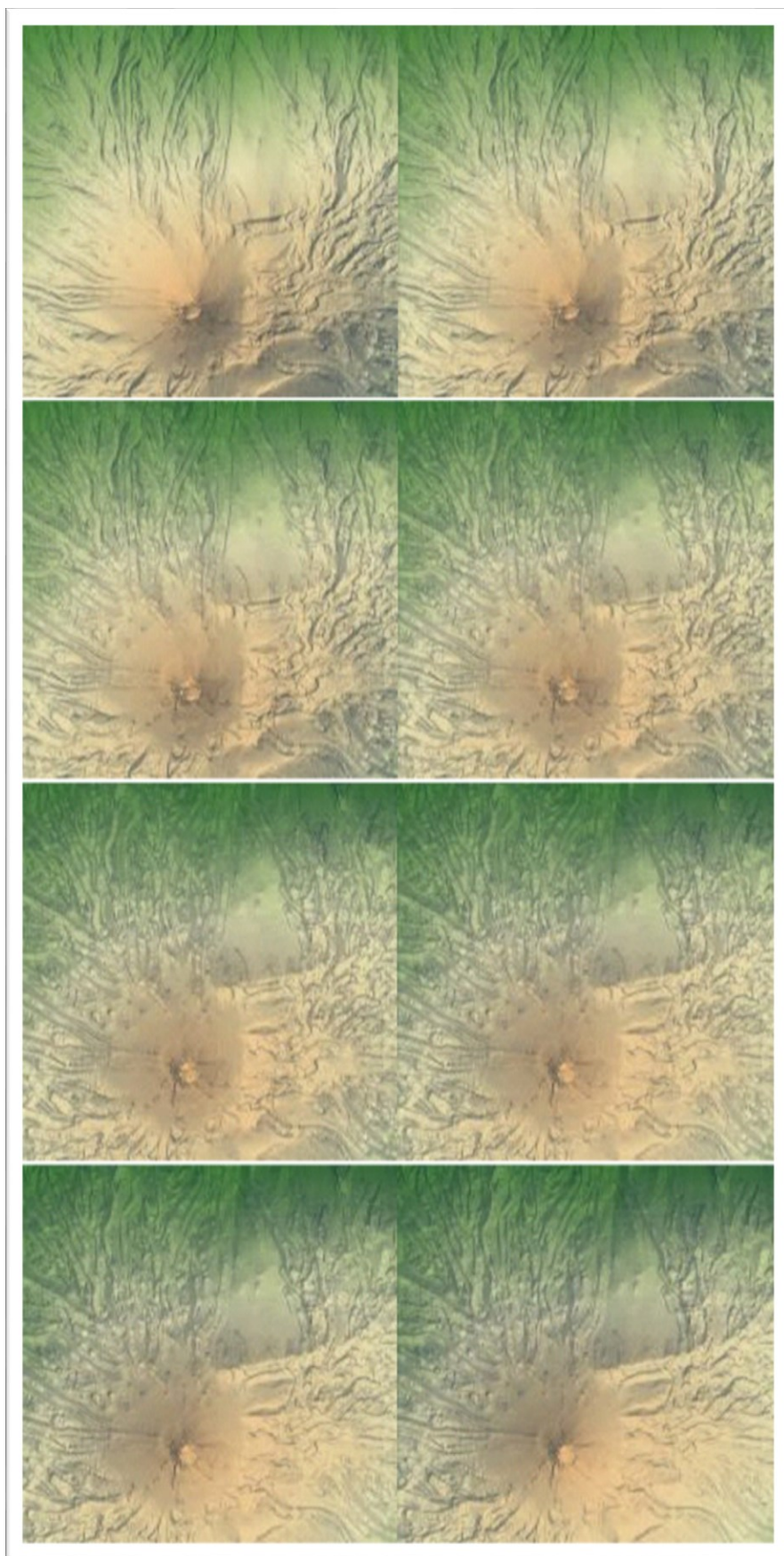


Figura 12.5. Ejemplo de algunos de los archivos generados mediante la opción “Hillshades Multiparámetro” del software MEL
Fuente: elaboración propia





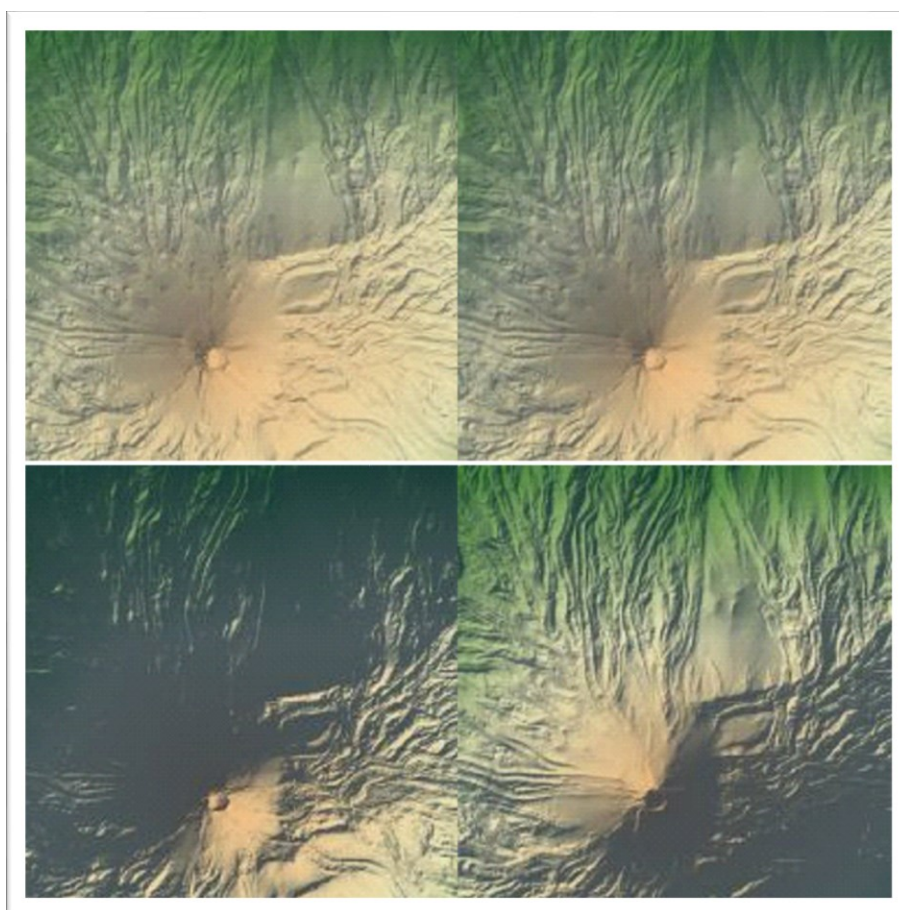


Figura 12.6. Ejemplo de algunos de los archivos generados mediante la opción “Mapas Hipsométricos Multialtitud” del software MEL
Fuente: elaboración propia

12.3. Anexo III: Generación automatizada de mapas hipsométricos

En este anexo se muestran ejemplos de casos de uso habituales en el trabajo cartográfico, como son la generación de mapas hipsométricos. El software de escritorio (MEL), permite el procesamiento masivo de datos, no sólo en formato LiDAR, sino también en otros formatos como el GeoTiff o el ASC. Gracias a la capacidad de trabajar también con archivos ASC, se han podido procesar los MDT25 del IGN y generar los mapas hipsométricos simplemente seleccionando los archivos que se querían procesar y eligiendo la paleta para asignar los colores en función de la elevación.

12.3.1. Santa Cruz de la Palma (Las Palmas de Gran Canaria)

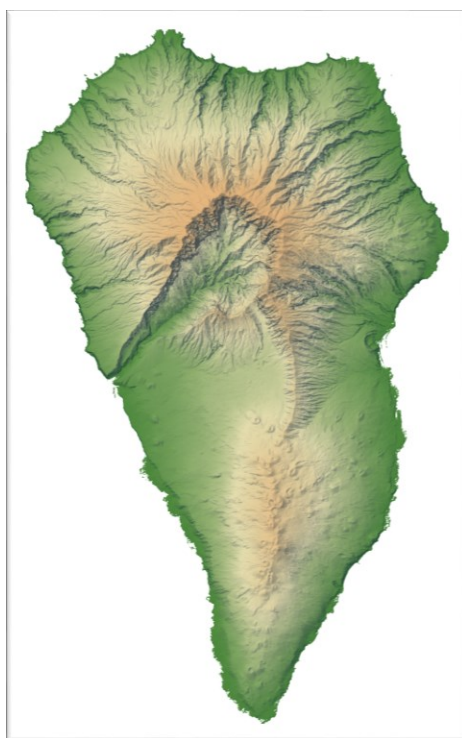


Figura 12.7. Mapa hipsométrico de Santa Cruz de la Palma, generado con el software MEL a partir del archivo .asc del MTN25 del IGN (paleta: atlas)
Fuente: elaboración propia, a partir de datos del IGN

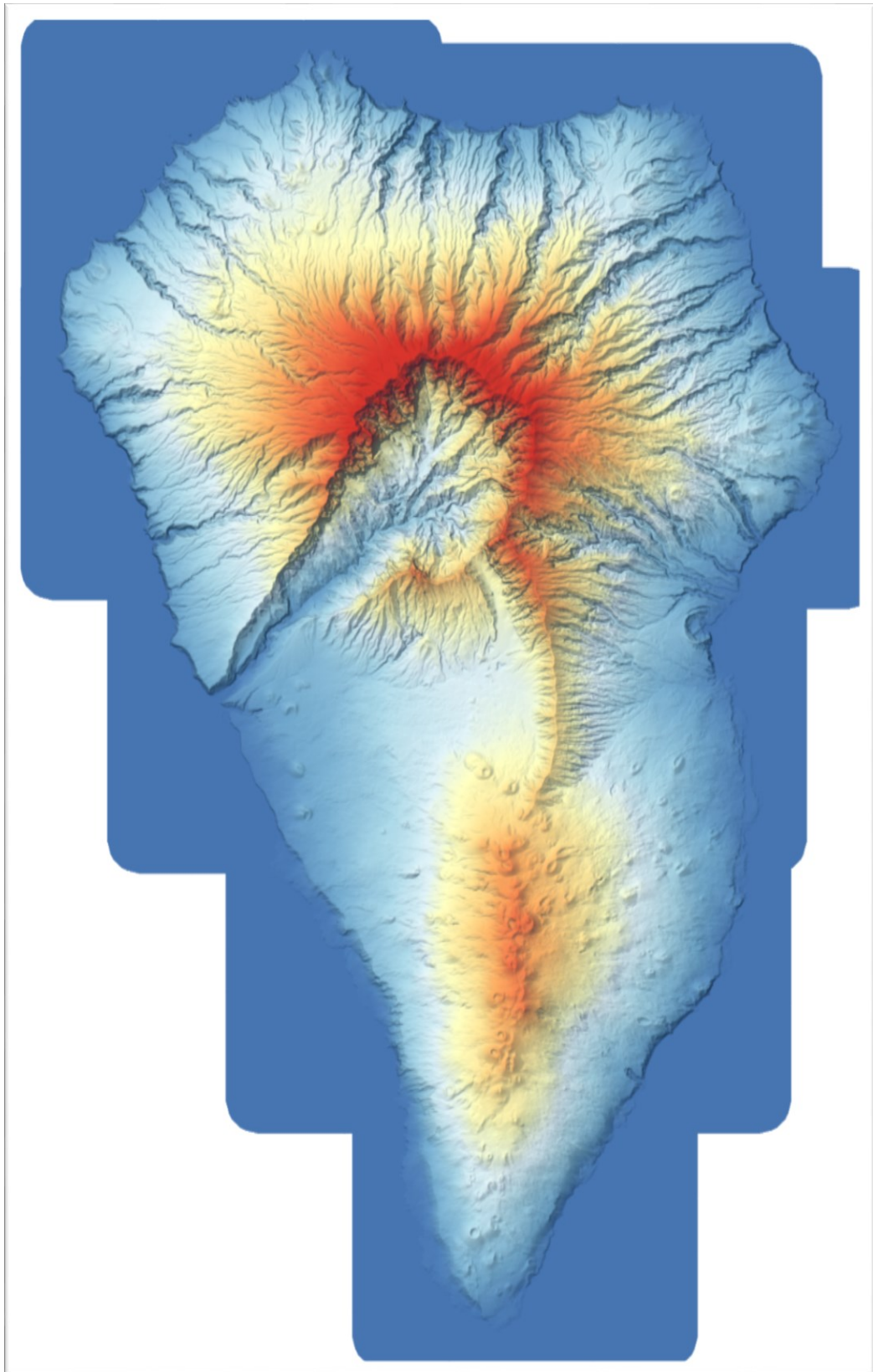


Figura 12.8. Mapa hipsométrico de Santa Cruz de la Palma, generado con el software MEL a partir de archivo .asc del MTN25 del IGN (paleta: muted)
Fuente: elaboración propia, a partir de datos del IGN

12.3.2. Comunidad de Madrid

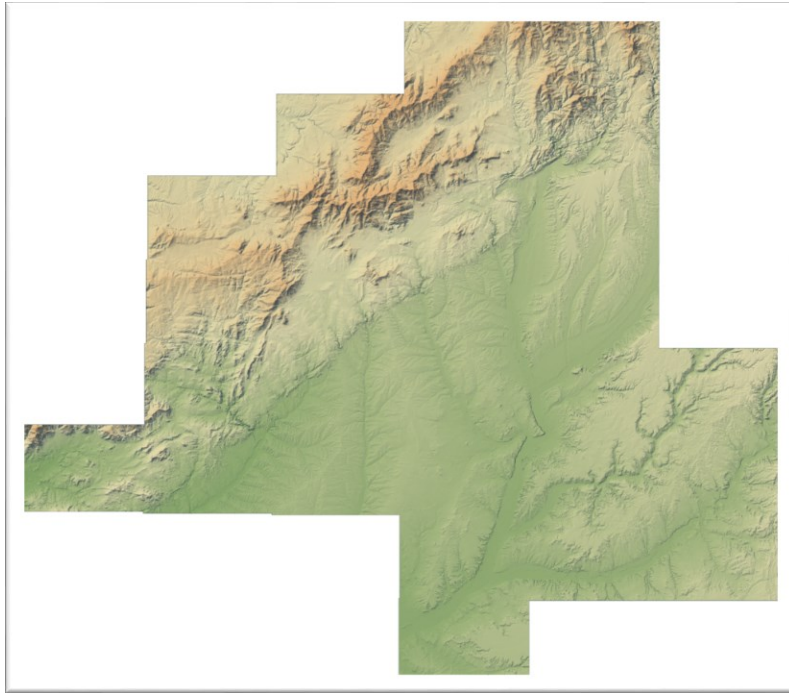


Figura 12.9. Mapa hipsométrico de la Comunidad de Madrid, generado con el software MEL a partir de archivo .asc del MTN25 del IGN (paleta: atlas)
Fuente: elaboración propia, a partir de datos del IGN

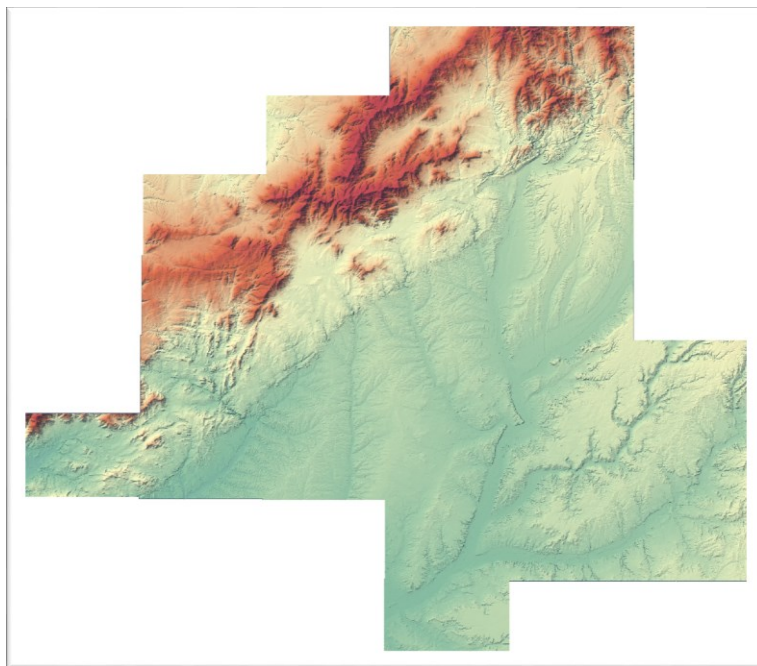


Figura 12.10. Mapa hipsométrico de la Comunidad de Madrid, generado con el software MEL a partir de archivos .asc del MTN25 del IGN (paleta: muted)
Fuente: elaboración propia, a partir de datos del IGN

13. Bibliografía

- Adelson, E. H., & Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(2), 284. doi:10.1364/josaa.2.000284
- Aggarwal, C. C. (2015). *Data classification: Algorithms and applications*. CRC Press, Taylor & Francis Group.
- Agrawal, P., Carreira, J., & Malik, J. (2015). Learning to See by Moving. 2015 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.13
- Aharon, M., Elad, M., & Bruckstein, A. (2006). K -SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11), 4311-4322. doi:10.1109/tsp.2006.881199
- Aigner, W., Miksch, S., Muller, W., Schumann, H., & Tominski, C. (2008, 01). Visual Methods for Analyzing Time-Oriented Data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1), 47-60 doi:10.1109/tvcg.2007.70415
- Alder, J., Hostetler, S., & Williams, D. (2013, 05). An Interactive Web Application for Visualizing Climate Data. *Eos, Transactions American Geophysical Union*, 94(22), 197-198. doi:10.1002/2013eo220001
- Ali, Z., & Gilania, S. (2013). Colorization by Example Using Dual-tree Complex Wavelet Transform and Jseg. *AASRI Procedia*, 4, 261-267. doi:10.1016/j.aasri.2013.10.039

- Amirian, P., Basiri, A., & Winstanley, A. (2014). Evaluation of Data Management Systems for Geospatial Big Data. Computational Science and Its Applications – ICCSA 2014 Lecture Notes in Computer Science, 678-690. doi:10.1007/978-3-319-09156-3_47
- An, X., & Pellacini, F. (2008). AppProp. ACM Transactions on Graphics, 27(3), 1-9. doi:10.1145/1360612.1360639
- Asai, K. (2010, 01). Visualization Based on Geographic Information in Augmented Reality. Augmented Reality. doi:10.5772/7134
- Ashikhmin, M. (2001). Synthesizing natural textures. Proceedings of the 2001 Symposium on Interactive 3D Graphics - SI3D '01. doi:10.1145/364338.364405
- Ashikhmin, M. (2003). Fast texture transfer. IEEE Computer Graphics and Applications, 23(4), 38-43. doi:10.1109/mcg.2003.1210863
- Auroux, D., & Masmoudi, M. (2006). A one-shot inpainting algorithm based on the topological asymptotic analysis. Clinics, 25(2-3). doi:10.1590/s1807-03022006000200008
- Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2481-2495. doi:10.1109/tpami.2016.2644615
- Balinsky, A., & Mohammad, N. (2010). Sparse natural image statistics and their applications to colorization and compression. 2010 IEEE International Conference on Image Processing. doi:10.1109/icip.2010.5652168

- Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., & Verdera, J. (2001). Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8), 1200-1211. doi:10.1109/83.935036
- Baluyan, H., Joshi, B., Hinai, A. A., & Woon, W. L. (2013). Novel Approach for Rooftop Detection Using Support Vector Machine. *ISRN Machine Vision*, 2013, 1-11. doi:10.1155/2013/819768
- Barnes, C., Goldman, D. B., Shechtman, E., & Finkelstein, A. (2011). The PatchMatch randomized matching algorithm for image manipulation. *Communications of the ACM*, 54(11), 103-110. doi:10.1145/2018396.2018421
- Barnes, C., Shechtman, E., Finkelstein, A., & Goldman, D. B. (2009). PatchMatch. *ACM Transactions on Graphics*, 28(3), 1-11. doi:10.1145/1531326.1531330
- Barron, J. T., & Poole, B. (2016). The Fast Bilateral Solver. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 617-632. doi:10.1007/978-3-319-46487-9_38
- Bashmal, L., Bazi, Y., Alhichri, H., Alrahal, M., Ammour, N., & Alajlan, N. (2018). Siamese-GAN: Learning Invariant Representations for Aerial Vehicle Image Categorization. *Remote Sensing*, 10(3), 351. doi:10.3390/rs10020351
- Basu, S., Ganguly, S., Mukhopadhyay, S., Dibiano, R., Karki, M., & Nemani, R. (2015). DeepSat. *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '15*. doi:10.1145/2820783.2820816

- Basu, S., Ganguly, S., Nemani, R. R., Mukhopadhyay, S., Zhang, G., Milesi, C., Li, S. (2015). A Semiautomated Probabilistic Framework for Tree-Cover Delineation From 1-m NAIP Imagery Using a High-Performance Computing Architecture. *IEEE Transactions on Geoscience and Remote Sensing*, 53(10), 5690-5708. doi:10.1109/tgrs.2015.2428197
- Basu, S., Karki, M., Ganguly, S., Dibiano, R., Mukhopadhyay, S., Gayaka, S., Nemani, R. (2016). Learning Sparse Feature Representations Using Probabilistic Quadtrees and Deep Belief Nets. *Neural Processing Letters*, 45(3), 855-867. doi:10.1007/s11063-016-9556-4
- Basu, S., Karki, M., Mukhopadhyay, S., Ganguly, S., Nemani, R., Dibiano, R., & Gayaka, S. (2016). A theoretical analysis of Deep Neural Networks for texture classification. 2016 International Joint Conference on Neural Networks (IJCNN). doi:10.1109/ijcnn.2016.7727306
- Beall, J. (2013, 12). The Open-Access Movement is Not Really about Open Access. *TripleC: Communication, Capitalism & Critique. Open Access Journal for a Global Sustainable Information Society*, 11(2), 589-597. doi:10.31269/vol11iss2pp589-597
- Bearman, A., Russakovsky, O., Ferrari, V., & Fei-Fei, L. (2016). What's the Point: Semantic Segmentation with Point Supervision. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 549-565. doi:10.1007/978-3-319-46478-7_34
- Bell, S., Upchurch, P., Snavely, N., & Bala, K. (2015). Material recognition in the wild with the Materials in Context Database. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2015.7298970
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798-1828. doi:10.1109/tpami.2013.50

- Bengio, Y., Goodfellow, I., & Courville, A. (2017). Deep learning. MIT Press.
- Bertalmio, M., Sapiro, G., Caselles, V., & Ballester, C. (2000). Image inpainting. Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '00. doi:10.1145/344779.344972
- Bertalmio, M., Vese, L., Sapiro, G., & Osher, S. (2003). Simultaneous structure and texture image inpainting. IEEE Transactions on Image Processing, 12(8), 882-889. doi:10.1109/tip.2003.815261
- Biharmonic equation. (2020, April 05). Retrieved from https://en.wikipedia.org/wiki/Biharmonic_equation
- Bittner, K., & Korner, M. (2018). Automatic Large-Scale 3D Building Shape Refinement Using Conditional Generative Adversarial Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). doi:10.1109/cvprw.2018.00249
- Blasch, E., Zheng, Y., & Liu, Z. (n.d.). Image Fusion Methods. Multispectral Image Fusion and Colorization. doi:10.1117/3.2316455.ch7
- Boden, A., & Avram, G. (2009). Bridging knowledge distribution - The role of knowledge brokers in distributed software development teams. 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering. doi:10.1109/chase.2009.5071402
- Boulos, M., Warren, J., Gong, J., & Yue, P. (2010). Web GIS in practice VIII: HTML5 and the canvas element for interactive online mapping. International Journal of Health Geographics, 9(1), 14. doi:10.1186/1476-072x-9-14

- Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., & Krishnan, D. (2017). Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.18
- Brewer, C. A. (1997, 01). Spectral Schemes: Controversial Color Use on Maps. *Cartography and Geographic Information Systems*, 24(4), 203-220 doi:10.1559/152304097782439231
- Buades, A., Coll, B., & Morel, J. (2011). Non-Local Means Denoising. *Image Processing On Line*, 1. doi:10.5201/ipol.2011.bcm_nlm
- Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., . . . Dicarlo, J. J. (2014). Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition. *PLoS Computational Biology*, 10(12). doi:10.1371/journal.pcbi.1003963
- Camps-Valls, G. (2009). Machine learning in remote sensing data processing. 2009 IEEE International Workshop on Machine Learning for Signal Processing. doi:10.1109/mlsp.2009.5306233
- Castronova, A. M., Goodall, J. L., & Elag, M. M. (2013, 03). Models as web services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard. *Environmental Modelling & Software*, 41, 72-83. doi:10.1016/j.envsoft.2012.11.010
- Chang, C., Chen, T., & Chung, P. (2018). Semi-supervised Learning Using Generative Adversarial Networks. 2018 IEEE Symposium Series on Computational Intelligence (SSCI). doi:10.1109/ssci.2018.8628663
- Chang, H., Fried, O., Liu, Y., Diverdi, S., & Finkelstein, A. (2015). Palette-based photo recoloring. *ACM Transactions on Graphics*, 34(4), 1-11. doi:10.1145/2766978

- Charpiat, G., Hofmann, M., & Schölkopf, B. (2008). Automatic Image Colorization Via Multimodal Predictions. *Lecture Notes in Computer Science Computer Vision – ECCV 2008*, 126-139. doi:10.1007/978-3-540-88690-7_10
- Chauhan, D., & Sharma, S. (2014). A Survey on Next Generation Internet Protocol: IPv6. *International Journal of Electronics and Electrical Engineering*, 143-146. doi:10.12720/ijeee.2.2.143-146
- Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 834-848. doi:10.1109/tpami.2017.2699184
- Chen, L., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *Computer Vision – ECCV 2018 Lecture Notes in Computer Science*, 833-851. doi:10.1007/978-3-030-01234-2_49
- Chen, X., Zhao, H., Li, P., & Yin, Z. (2006). Remote sensing image-based analysis of the relationship between urban heat island and land use/cover changes. *Remote Sensing of Environment*, 104(2), 133-146. doi:10.1016/j.rse.2005.11.016
- Chen, X., Zou, D., Zhao, Q., & Tan, P. (2012). Manifold preserving edit propagation. *ACM Transactions on Graphics*, 31(6), 1-7. doi:10.1145/2366145.2366151
- Chen, Y., Li, W., & Gool, L. V. (2018). ROAD: Reality Oriented Adaptation for Semantic Segmentation of Urban Scenes. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. doi:10.1109/cvpr.2018.00823

- Cheng, L., & Vishwanathan, S. V. (2007). Learning to compress images and videos. Proceedings of the 24th International Conference on Machine Learning - ICML '07. doi:10.1145/1273496.1273517
- Cheng, Z., Yang, Q., & Sheng, B. (2015). Deep Colorization. 2015 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.55
- Chia, A. Y., Zhuo, S., Gupta, R. K., Tai, Y., Cho, S., Tan, P., & Lin, S. (2011). Semantic colorization with internet images. Proceedings of the 2011 SIGGRAPH Asia Conference on - SA '11. doi:10.1145/2024156.2024190
- Chui, C. K., & Mhaskar, H. (2010). MRA contextual-recovery extension of smooth functions on manifolds. Applied and Computational Harmonic Analysis, 28(1), 104-113. doi:10.1016/j.acha.2009.04.004
- Chuvieco, E., & Huete, A. (2010). Fundamentals of satellite remote sensing. CRC Press.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.350
- Crampton, J. W. (2002, 01). Interactivity Types in Geographic Visualization. Cartography and Geographic Information Science, 29(2), 85-98. doi:10.1559/152304002782053314
- Criminisi, A., Perez, P., & Toyama, K. (2004). Region Filling and Object Removal by Exemplar-Based Image Inpainting. IEEE Transactions on Image Processing, 13(9), 1200-1212. doi:10.1109/tip.2004.833105

- Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable Convolutional Networks. 2017 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2017.89
- Damelin, S. B., & Hoang, N. S. (2018). On Surface Completion and Image Inpainting by Biharmonic Functions: Numerical Aspects. International Journal of Mathematics and Mathematical Sciences, 2018, 1-8. doi:10.1155/2018/3950312
- Deshpande, A., Rock, J., & Forsyth, D. (2015). Learning Large-Scale Automatic Image Colorization. 2015 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.72
- Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised Visual Representation Learning by Context Prediction. 2015 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.167
- Dong, H., Yu, S., Wu, C., & Guo, Y. (2017). Semantic Image Synthesis via Adversarial Learning. 2017 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2017.608
- Dong, L., & Shan, J. (2013). A comprehensive review of earthquake-induced building damage detection with remote sensing techniques. ISPRS Journal of Photogrammetry and Remote Sensing, 84, 85-99. doi:10.1016/j.isprsjprs.2013.06.011
- Dove, E. S. (2015, 12). Reflections on the Concept of Open Data. SCRIPTed, 12(2), 154-166. doi:10.2966/script.120215.154
- Drori, I., Cohen-Or, D., & Yeshurun, H. (2003). Fragment-based image completion. ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03. doi:10.1145/1201775.882267

- Dykes, J., Maceachren, A. M., & Kraak, M. (2005). Introduction Exploring Geovisualization. *Exploring Geovisualization*, 1-19. doi:10.1016/b978-008044531-1/50419-x
- Döllner, J. (2005). Geovisualization and Real-Time 3D Computer Graphics. *Exploring Geovisualization*, 325-343. doi:10.1016/b978-008044531-1/50434-6
- Efros, A. A., & Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '01*. doi:10.1145/383259.383296
- Efros, A., & Leung, T. (1999). Texture synthesis by non-parametric sampling. *Proceedings of the Seventh IEEE International Conference on Computer Vision*. doi:10.1109/iccv.1999.790383
- Elad, M., Starck, J., Querre, P., & Donoho, D. (2005). Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis*, 19(3), 340-358. doi:10.1016/j.acha.2005.03.005
- Elgammal, A., & Lee, C. (n.d.). Separating style and content on a nonlinear manifold. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. doi:10.1109/cvpr.2004.1315070
- Endo, Y., Iizuka, S., Kanamori, Y., & Mitani, J. (2016). DeepProp: Extracting Deep Features from a Single Image for Edit Propagation. *Computer Graphics Forum*, 35(2), 189-201. doi:10.1111/cgf.12822
- Esedoglu, S., & Shen, J. (2002). Digital inpainting based on the Mumford–Shah–Euler image model. *European Journal of Applied Mathematics*, 13(04). doi:10.1017/s0956792502004904

- Esedoglu, S., Tsai, R., & Ruuth, S. (2008). Threshold dynamics for high order geometric motions. *Interfaces and Free Boundaries*, 263-282. doi:10.4171/ifb/189
- Evans, J. A., & Reimer, J. (2009, 02). Open Access and Global Participation in Science. *Science*, 323(5917), 1025-1025. doi:10.1126/science.1154562
- Evans, M., Oliver, D., Zhou, X., & Shekhar, S. (2014, 01). Spatial Big Data. *Big Data*, 149-176. doi:10.1201/b16524-9
- Everingham, M., Gool, L. V., Williams, C. K., Winn, J., & Zisserman, A. (2009). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2), 303-338. doi:10.1007/s11263-009-0275-4
- Farabet, C., Couprie, C., Najman, L., & Lecun, Y. (2013). Learning Hierarchical Features for Scene Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1915-1929. doi:10.1109/tpami.2012.231
- Ferrer-Sapena, A., Peset, F., & Aleixandre-Benavent, R. (2011, 05). Acceso a Los Datos Públicos y Su Reutilización: Open Data y Open Government. *El Profesional De La Informacion*, 20(3), 260-269. doi:10.3145/epi.2011.may.03
- Frye, R., & Mckenney, M. (2014, 07). Big Data Storage Techniques for Spatial Databases: Implications of Big Data Architecture on Spatial Query Processing. *Studies in Big Data Information Granularity, Big Data, and Computational Intelligence*, 297-323. doi:10.1007/978-3-319-08254-7_14

- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Lempitsky, V. (2017). Domain-Adversarial Training of Neural Networks. *Domain Adaptation in Computer Vision Applications Advances in Computer Vision and Pattern Recognition*, 189-209. doi:10.1007/978-3-319-58347-1_10
- Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.265
- Gatys, L., Ecker, A., & Bethge, M. (2016). A Neural Algorithm of Artistic Style. *Journal of Vision*, 16(12), 326. doi:10.1167/16.12.326
- Gharbi, M., Chaurasia, G., Paris, S., & Durand, F. (2016). Deep joint demosaicking and denoising. *ACM Transactions on Graphics*, 35(6), 1-12. doi:10.1145/2980179.2982399
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2014.81
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (pp. 2672–2680). MIT Press. ArXiv, abs/1406.2661
- Guadarrama, S., Dahl, R., Bieber, D., Shlens, J., Norouzi, M., & Murphy, K. (2017). PixColor: Pixel Recursive Colorization. *Proceedings of the British Machine Vision Conference 2017*. doi:10.5244/c.31.112
- Guclu, U., & Gerven, M. A. (2015). Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral

Stream. Journal of Neuroscience, 35(27), 10005-10014.
doi:10.1523/jneurosci.5023-14.2015

- Gupta, R. K., Chia, A. Y., Rajan, D., Ng, E. S., & Zhiyong, H. (2012). Image colorization using similar images. Proceedings of the 20th ACM International Conference on Multimedia - MM '12. doi:10.1145/2393347.2393402
- Gyrard, A. (2013). An Architecture to Aggregate Heterogeneous and Semantic Sensed Data. The Semantic Web: Semantics and Big Data Lecture Notes in Computer Science, 697-701. doi:10.1007/978-3-642-38288-8_54
- Hariharan, B., Arbelaez, P., Girshick, R., & Malik, J. (2015). Hypercolumns for object segmentation and fine-grained localization. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2015.7298642
- Harley, A. W., Derpanis, K. G., & Kokkinos, I. (2017). Segmentation-Aware Convolutional Networks Using Local Attention Masks. 2017 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2017.539
- Hays, J., & Efros, A. A. (2007). Scene completion using millions of photographs. ACM SIGGRAPH 2007 Papers on - SIGGRAPH '07. doi:10.1145/1275808.1276382
- Hays, J., & Efros, A. A. (2008). Scene completion using millions of photographs. Communications of the ACM, 51(10), 87-94. doi:10.1145/1400181.1400202
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015

- IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.123
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.90
 - He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity Mappings in Deep Residual Networks. Computer Vision – ECCV 2016 Lecture Notes in Computer Science, 630-645. doi:10.1007/978-3-319-46493-0_38
 - He, X., Ji, M., & Bao, H. (2009). A unified active and semi-supervised learning framework for image compression. 2009 IEEE Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2009.5206835
 - Heeger, D., & Bergen, J. (n.d.). Pyramid-based texture analysis/synthesis. Proceedings., International Conference on Image Processing. doi:10.1109/icip.1995.537718
 - Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., & Salesin, D. H. (2001). Image analogies. Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '01. doi:10.1145/383259.383295
 - Hu, W., & Qin, K. (2009). Fast Multi-Resolution Colorization of High-Resolution Gray Images. Chinese Journal of Computers, 32(5), 1062-1068. doi:10.3724/sp.j.1016.2009.01062
 - Huang, Y., Tung, Y., Chen, J., Wang, S., & Wu, J. (2005). An adaptive edge detection based colorization algorithm and its applications. Proceedings of the 13th Annual ACM International Conference on Multimedia - MULTIMEDIA '05. doi:10.1145/1101149.1101223

- Huber, P. J. (1964). Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1), 73-101. doi:10.1214/aoms/1177703732
- Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2016). Let there be color! *ACM Transactions on Graphics*, 35(4), 1-11. doi:10.1145/2897824.2925974
- Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36(4), 1-14. doi:10.1145/3072959.3073659
- Inpainting. (2020, June 28). Retrieved from <https://en.wikipedia.org/wiki/Inpainting>
- Iosifescu-Enescu, I., Hugentobler, M., & Hurni, L. (2010, 09). Web cartography with open standards – A solution to cartographic challenges of environmental management. *Environmental Modelling & Software*, 25(9), 988-999. doi:10.1016/j.envsoft.2009.10.017
- Isola, P., Zhu, J., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.632
- Jayaraman, D., & Grauman, K. (2015). Learning Image Representations Tied to Ego-Motion. 2015 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.166
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Darrell, T. (2014). Caffe. *Proceedings of the ACM International Conference on Multimedia - MM '14*. doi:10.1145/2647868.2654889

- Jin, X., & Davis, C. H. (2005). Automated Building Extraction from High-Resolution Satellite Imagery in Urban Areas Using Structural, Contextual, and Spectral Information. *EURASIP Journal on Advances in Signal Processing*, 2005(14). doi:10.1155/asp.2005.2196
- Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 694-711. doi:10.1007/978-3-319-46475-6_43
- Joshi, B., Baluyan, H., Hinai, A. A., & Woon, W. L. (2014). Automatic Rooftop Detection Using a Two-Stage Classification. *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. doi:10.1109/uksim.2014.89
- Kamburugamuve, S., Ekanayake, S., Pathirage, M., & Fox, G. (2016, 05). Towards High Performance Processing of Streaming Data in Large Data Centers. *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. doi:10.1109/ipdpsw.2016.103
- Kan, M., Shan, S., Chang, H., & Chen, X. (2014). Stacked Progressive Auto-Encoders (SPA-E) for Face Recognition Across Poses. *2014 IEEE Conference on Computer Vision and Pattern Recognition*. doi:10.1109/cvpr.2014.243
- Kaneko, T., Hiramatsu, K., & Kashino, K. (2017). Generative Attribute Controller with Conditional Filtered Generative Adversarial Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2017.741
- Karayev, S., Hertzmann, A., Trentacoste, M., Han, H., Winnemoeller, H., Agarwala, A., & Darrell, T. (2014). Recognizing Image Style. *Proceedings of the British Machine Vision Conference 2014*. doi:10.5244/c.28.122

- Karki, M., Dibiano, R., Basu, S., & Mukhopadhyay, S. (2017). Core Sampling Framework for Pixel Classification. *Artificial Neural Networks and Machine Learning – ICANN 2017 Lecture Notes in Computer Science*, 617-625. doi:10.1007/978-3-319-68612-7_70
- Kemp, Z. (2005). A Knowledge-based Collaborative Environment for Geovisualization. *Exploring Geovisualization*, 495-512. doi:10.1016/b978-008044531-1/50443-7
- Khaligh-Razavi, S., & Kriegeskorte, N. (2014). Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. *PLoS Computational Biology*, 10(11). doi:10.1371/journal.pcbi.1003915
- Kiani, L., Saeed, M., & Nezamabadi-Pour, H. (2020). Image Colorization Using Generative Adversarial Networks and Transfer Learning. 2020 International Conference on Machine Vision and Image Processing (MVIP). doi:10.1109/mvip49855.2020.9116882
- Kim, D., Jung, H., Lee, J., & Kim, J. (2018). Improved InfoGAN: Generating High Quality Images with Learning Disentangled Representation. *Robot Intelligence Technology and Applications 5 Advances in Intelligent Systems and Computing*, 43-48. doi:10.1007/978-3-319-78452-6_5
- Kirschenbauer, S. (2005). Applying “True 3D” Techniques to Geovisualization. *Exploring Geovisualization*, 363-387. doi:10.1016/b978-008044531-1/50436-x
- Kohlhammer, J., May, T., & Hoffmann, M. (2009). Visual Analytics for the Strategic Decision Making Process. *GeoSpatial Visual Analytics NATO Science for Peace and Security Series C: Environmental Security*, 299-310. doi:10.1007/978-90-481-2899-0_23
- Kokaram, A. C. (1998). Motion Picture Restoration. doi:10.1007/978-1-4471-3485-5

- Komodakis, N., & Tziritas, G. (2007). Image Completion Using Efficient Belief Propagation Via Priority Scheduling and Dynamic Pruning. *IEEE Transactions on Image Processing*, 16(11), 2649-2661. doi:10.1109/tip.2007.906269
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90. doi:10.1145/3065386
- Kwatra, V., Essa, I., Bobick, A., & Kwatra, N. (2005). Texture optimization for example-based synthesis. *ACM SIGGRAPH 2005 Papers on - SIGGRAPH '05*. doi:10.1145/1186822.1073263
- Kyprianidis, J. E., Collomosse, J., Wang, T., & Isenberg, T. (2013). State of the "Art": A Taxonomy of Artistic Stylization Techniques for Images and Video. *IEEE Transactions on Visualization and Computer Graphics*, 19(5), 866-885. doi:10.1109/tvcg.2012.160
- Larsson, G., Maire, M., & Shakhnarovich, G. (2016). Learning Representations for Automatic Colorization. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 577-593. doi:10.1007/978-3-319-46493-0_35
- Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324. doi:10.1109/5.726791
- Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Shi, W. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2017.19
- Lee, H., Seo, S., & Yoon, K. (2011). Directional texture transfer with edge enhancement. *Computers & Graphics*, 35(1), 81-91. doi:10.1016/j.cag.2010.11.008

- Leissler, M., Hemmje, M., & Neuhold, E. (2000). Automatic Updates of Interactive Information Visualization User Interfaces through Database Triggers. *Advances in Visual Information Management*, 341-366. doi:10.1007/978-0-387-35504-7_22
- Levin, A., Lischinski, D., & Weiss, Y. (2004). Colorization using optimization. *ACM Transactions on Graphics*, 23(3), 689-694. doi:10.1145/1015706.1015780
- Li, B., Lai, Y., & Rosin, P. L. (2017). Example-based image colorization via automatic feature selection and fusion. *Neurocomputing*, 266, 687-698. doi:10.1016/j.neucom.2017.05.083
- Li, X., Zhao, H., Nie, G., & Huang, H. (2015). Image recoloring using geodesic distance based color harmonization. *Computational Visual Media*, 1(2), 143-155. doi:10.1007/s41095-015-0013-5
- Li, Y., Adelson, E., & Agarwala, A. (2008). ScribbleBoost: Adding Classification to Edge-Aware Interpolation of Local Image and Video Adjustments. *Computer Graphics Forum*, 27(4), 1255-1264. doi:10.1111/j.1467-8659.2008.01264.x
- Li, Y., Liu, S., Yang, J., & Yang, M. (2017). Generative Face Completion. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.624
- Liao, Z., Yu, Y., & Chen, B. (2010, 10). Anomaly detection in GPS data based on visual analytics. 2010 IEEE Symposium on Visual Analytics Science and Technology. doi:10.1109/vast.2010.5652467
- Lim, B., Son, S., Kim, H., Nah, S., & Lee, K. M. (2017). Enhanced Deep Residual Networks for Single Image Super-Resolution. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). doi:10.1109/cvprw.2017.151

- Liu, X., Wan, L., Qu, Y., Wong, T., Lin, S., Leung, C., & Heng, P. (2008). Intrinsic colorization. ACM SIGGRAPH Asia 2008 Papers on - SIGGRAPH Asia '08. doi:10.1145/1457515.1409105
- Liu, Y., Cohen, M., Uyttendaele, M., & Rusinkiewicz, S. (2014). AutoStyle: Automatic Style Transfer from Image Collections to Users' Images. Computer Graphics Forum, 33(4), 21-31. doi:10.1111/cgf.12409
- Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep Learning Face Attributes in the Wild. 2015 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.425
- Lonergan, C., & Hedley, N. (2014, 09). Flexible Mixed Reality and Situated Simulation as Emerging Forms of Geovisualization. Cartographica: The International Journal for Geographic Information and Geovisualization, 49(3), 175-187. doi:10.3138/carto.49.3.2440
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2015.7298965
- Luna-Reyes, L. F. (2018). US open data policy. Proceedings of the 19th Annual International Conference on Digital Government Research Governance in the Data Age - Dgo '18. doi:10.1145/3209281.3209308
- Mahendran, A., & Vedaldi, A. (2015). Understanding deep image representations by inverting them. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2015.7299155
- Mairal, J., Elad, M., & Sapiro, G. (2008). Sparse Representation for Color Image Restoration. IEEE Transactions on Image Processing, 17(1), 53-69. doi:10.1109/tip.2007.911828
- Marsland, S. (2014). "Machine Learning: An Algorithmic Perspective, Second Edition".

- Masnou, S., & Morel, J. (n.d.). Level lines based disocclusion. Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269). doi:10.1109/icip.1998.999016
- Meehl, G. A. (2004). More Intense, More Frequent, and Longer Lasting Heat Waves in the 21st Century. *Science*, 305(5686), 994-997. doi:10.1126/science.1098704
- Monino, J., & Sedkaoui, S. (2016, 03). The Big Data Revolution. *Big Data, Open Data and Data Development*, 1-21. doi:10.1002/9781119285199.ch1
- Moreno Serrano, F. (1991). Superficies de erosión y tectónica neógena en el extremo occidental del Sistema central español. *Geogaceta*, 9: 47-50.
- Morimoto, Y., Taguchi, Y., & Naemura, T. (2009). Automatic colorization of grayscale images using multiple images on the web. SIGGRAPH 2009: Talks on - SIGGRAPH '09. doi:10.1145/1597990.1598049
- Muehlenhaus, I. (2014). *Web cartography: Map design for interactive and mobile devices*. CRC Press.
- Niranjan, M. (1999). Support vector machines: A tutorial overview and critical appraisal. IEE Colloquium on Applied Statistical Pattern Recognition. doi:10.1049/ic:19990359
- Noroozi, M., & Favaro, P. (2016). Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 69-84. doi:10.1007/978-3-319-46466-4_5
- Orford, S. (n.d.). Visualization with High-Resolution Aerial Photography in Planning-Related Property Research. *Geographic Visualization*, 141-158. doi:10.1002/9780470987643.ch7

- Owens, A., Isola, P., Mcdermott, J., Torralba, A., Adelson, E. H., & Freeman, W. T. (2016). Visually Indicated Sounds. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.264
- PHP - A Server-Side Scripting Language. (2011). Web Application Design and Implementation. doi:10.1109/9780470083963.ch5
- Patel, V. M., Gopalan, R., Li, R., & Chellappa, R. (2015). Visual Domain Adaptation: A survey of recent advances. IEEE Signal Processing Magazine, 32(3), 53-69. doi:10.1109/msp.2014.2347059
- Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context Encoders: Feature Learning by Inpainting. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.278
- Patterson, G., & Hays, J. (2012). SUN attribute database: Discovering, annotating, and recognizing scene attributes. 2012 IEEE Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2012.6247998
- Patwardhan, K. A., Sapiro, G., & Bertalmio, M. (2007). Video Inpainting Under Constrained Camera Motion. IEEE Transactions on Image Processing, 16(2), 545-553. doi:10.1109/tip.2006.888343
- Peiravian, F., Kermanshah, A., & Derrible, S. (2014, 10). Spatial data analysis of complex urban systems. 2014 IEEE International Conference on Big Data (Big Data). doi:10.1109/bigdata.2014.7004405
- Peng, X., & Saenko, K. (2018). Synthetic to Real Adaptation with Generative Correlation Alignment Networks. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). doi:10.1109/wacv.2018.00219

- Pohl, C., & L., V. G. (2017). Remote sensing image fusion: A practical guide. Taylor & Francis.
- Popescu, S. (2010). LiDAR: Remote sensing of terrestrial environments. CRC.
- Pérez, P., Gangnet, M., & Blake, A. (2003). Poisson image editing. ACM Transactions on Graphics, 22(3), 313-318. doi:10.1145/882262.882269
- Raad, L., & Galerne, B. (2017). Efros and Freeman Image Quilting Algorithm for Texture Synthesis. Image Processing On Line, 7, 1-22. doi:10.5201/ipol.2017.171
- Ramanarayanan, G., Ferwerda, J., Walter, B., & Bala, K. (2007). Visual equivalence. ACM SIGGRAPH 2007 Papers on - SIGGRAPH '07. doi:10.1145/1275808.1276472
- Ratliff, N. D., Silver, D., & Bagnell, J. A. (2009). Learning to search: Functional gradient techniques for imitation learning. Autonomous Robots, 27(1), 25-53. doi:10.1007/s10514-009-9121-3
- Reagan, J. A. (1991). LiDAR. Society of Photo-Optical Instrumentation Engineers.
- Reddy, M. (n.d.). Three-Dimensional Visualization. Encyclopedia of Geographic Information Science. doi:10.4135/9781412953962.n212
- Reed, C. N. (n.d.). The Open Geospatial Consortium and Web Services Standards. Advances in Geospatial Technologies Geospatial Web Services, 1-16. doi:10.4018/978-1-60960-192-8.ch001
- Rees, G. (2001). Physical principles of remote sensing. Cambridge University Press.

- Ren, H., Li, J., & Gao, N. (2018). Automatic Sketch Colorization with Tandem Conditional Adversarial Networks. 2018 11th International Symposium on Computational Intelligence and Design (ISCID). doi:10.1109/iscid.2018.00009
- Ren, K., Sun, H., Jia, Q., & Shi, J. (2009). Building recognition from aerial images combining segmentation and shadow. 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems. doi:10.1109/icicisys.2009.5357616
- Richter, S. R., Vineet, V., Roth, S., & Koltun, V. (2016). Playing for Data: Ground Truth from Computer Games. Computer Vision – ECCV 2016 Lecture Notes in Computer Science, 102-118. doi:10.1007/978-3-319-46475-6_7
- Rodgers, P. (2005). Graph Drawing Techniques for Geographic Visualization. Exploring Geovisualization, 143-158. doi:10.1016/b978-008044531-1/50425-5
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Lecture Notes in Computer Science Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, 234-241. doi:10.1007/978-3-319-24574-4_28
- Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.352
- Rula, A., Maurino, A., & Batini, C. (2016). Data Quality Issues in Linked Open Data. Data-Centric Systems and Applications Data and Information Quality, 87-112. doi:10.1007/978-3-319-24106-7_4

- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211-252. doi:10.1007/s11263-015-0816-y
- Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting Visual Category Models to New Domains. *Computer Vision – ECCV 2010 Lecture Notes in Computer Science*, 213-226. doi:10.1007/978-3-642-15561-1_16
- Saleh, F., Aliakbarian, M. S., Salzmann, M., Petersson, L., Gould, S., & Alvarez, J. M. (2016). Built-in Foreground/Background Prior for Weakly-Supervised Semantic Segmentation. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 413-432. doi:10.1007/978-3-319-46484-8_25
- Salvaris, M., Dean, D., & Tok, W. H. (2018). Generative Adversarial Networks. *Deep Learning with Azure*, 187-208. doi:10.1007/978-1-4842-3679-6_8
- Sangkloy, P., Lu, J., Fang, C., Yu, F., & Hays, J. (2017). Scribbler: Controlling Deep Image Synthesis with Sketch and Color. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2017.723
- Schonfeld, E., Schiele, B., & Khoreva, A. (2020). A U-Net Based Discriminator for Generative Adversarial Networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr42600.2020.00823
- Selim, A., Elgharib, M., & Doyle, L. (2016). Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics*, 35(4), 1-18. doi:10.1145/2897824.2925968

- Shafaei, A., Little, J., & Schmidt, M. (2016). Play and Learn: Using Video Games to Train Computer Vision Models. *Proceedings of the British Machine Vision Conference 2016*. doi:10.5244/c.30.26
- Shamsabadi, A. S., Sanchez-Matilla, R., & Cavallaro, A. (2020). ColorFool: Semantic Adversarial Colorization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr42600.2020.00123
- Shen, J., Kang, S. H., & Chan, T. F. (2003). Euler's Elastica and Curvature-Based Inpainting. *SIAM Journal on Applied Mathematics*, 63(2), 564-592. doi:10.1137/s0036139901390088
- Shimoda, W., & Yanai, K. (2016). Distinct Class-Specific Saliency Maps for Weakly Supervised Semantic Segmentation. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 218-234. doi:10.1007/978-3-319-46493-0_14
- Shiratori, T., Matsushita, Y., Tang, X., & Kang, S. B. (n.d.). Video Completion by Motion Field Transfer. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*. doi:10.1109/cvpr.2006.330
- Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., & Webb, R. (2017). Learning from Simulated and Unsupervised Images through Adversarial Training. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2017.241
- Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor Segmentation and Support Inference from RGBD Images. *Computer Vision – ECCV 2012 Lecture Notes in Computer Science*, 746-760. doi:10.1007/978-3-642-33715-4_54

- Simakov, D., Caspi, Y., Shechtman, E., & Irani, M. (2008). Summarizing visual data using bidirectional similarity. 2008 IEEE Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2008.4587842
- Simoncelli, E., & Portilla, J. (n.d.). Texture characterization via joint statistics of wavelet coefficient magnitudes. Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269). doi:10.1109/icip.1998.723417
- Simonyan, K., & Zisserman, A. (2015, April 10). Very Deep Convolutional Networks for Large-Scale Image Recognition. Web: <http://arxiv.org/abs/1409.1556>
- Sohn, G., Jwa, Y., Jung, J., & Kim, H. (2012). An Implicit Regularization For 3D Building Rooftop Modeling Using Airborne LiDAR Data. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, I-3, 305-310. doi:10.5194/isprsannals-i-3-305-2012
- Song, X., Dai, Y., & Qin, X. (2017). Deep Depth Super-Resolution: Learning Depth Super-Resolution Using Deep Convolutional Neural Network. Computer Vision – ACCV 2016 Lecture Notes in Computer Science, 360-376. doi:10.1007/978-3-319-54190-7_22
- Song, X., Sexton, J. O., Huang, C., Channan, S., & Townshend, J. R. (2016). Characterizing the magnitude, timing, and duration of urban growth from time series of Landsat-based estimates of impervious cover. Remote Sensing of Environment, 175, 1-13. doi:10.1016/j.rse.2015.12.027
- Special issue on cyber-physical cloud systems. (2015, 11). IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45(11), 1477-1478 doi:10.1109/tsmc.2015.2489878

- Sukumar, S. R. (2015, 10). Open research challenges with Big Data — A data-scientist's perspective. 2015 IEEE International Conference on Big Data (Big Data). doi:10.1109/bigdata.2015.7363882
- Sun, J., Yuan, L., Jia, J., & Shum, H. (2005). Image completion with structure propagation. ACM SIGGRAPH 2005 Papers on - SIGGRAPH '05. doi:10.1145/1186822.1073274
- Sundaram, N., Brox, T., & Keutzer, K. (2010). Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow. Computer Vision – ECCV 2010 Lecture Notes in Computer Science, 438-451. doi:10.1007/978-3-642-15549-9_32
- Surhone, L. M. (2010). Visual analytics: Information visualization, scientific visualization, user interface, interactive visualization, computer graphics, intercation. Betascript Publ.
- Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. 2014 IEEE Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2014.220
- Tan, Y., & Shi, Y. (2016). Data mining and big data: First International Conference, DMBD 2016, Bali, Indonesia, June 25-30, 2016. Proceedings. Springer.
- Telea, A. (2004). An Image Inpainting Technique Based on the Fast Marching Method. Journal of Graphics Tools, 9(1), 23-34. doi:10.1080/10867651.2004.10487596
- Telea, A. (2004). An Image Inpainting Technique Based on the Fast Marching Method. Journal of Graphics Tools, 9(1), 23-34. doi:10.1080/10867651.2004.10487596

- The Top Ten Algorithms in Data Mining. (2009). doi:10.1201/9781420089653
- Tsai, Y., Hung, W., Schuler, S., Sohn, K., Yang, M., & Chandraker, M. (2018). Learning to Adapt Structured Output Space for Semantic Segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2018.00780
- Tschumperlé, D. (2006). Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's. International Journal of Computer Vision, 68(1), 65-82. doi:10.1007/s11263-006-5631-z
- Tzeng, E., Hoffman, J., Darrell, T., & Saenko, K. (2015). Simultaneous Deep Transfer Across Domains and Tasks. 2015 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.463
- Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., & Geiger, A. (2017). Sparsity Invariant CNNs. 2017 International Conference on 3D Vision (3DV). doi:10.1109/3dv.2017.00012
- Vandal, T., Kodra, E., Ganguly, S., Michaelis, A., Nemani, R., & Ganguly, A. R. (2017). DeepSD. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. doi:10.1145/3097983.3098004
- Vazquez, D., Lopez, A. M., Marin, J., Ponsa, D., & Geronimo, D. (2014). Virtual and Real World Adaptation for Pedestrian Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(4), 797-809. doi:10.1109/tpami.2013.163
- Wang, B., Yu, Y., Wong, T., Chen, C., & Xu, Y. (2010). Data-driven image color theme enhancement. ACM Transactions on Graphics, 29(6), 1-10. doi:10.1145/1882261.1866172

- Wang, T., Zhu, J., Hiroaki, E., Chandraker, M., Efros, A. A., & Ramamoorthi, R. (2016). A 4D Light-Field Dataset and CNN Architectures for Material Recognition. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 121-138. doi:10.1007/978-3-319-46487-9_8
- Wang, X., & Gupta, A. (2015). Unsupervised Learning of Visual Representations Using Videos. *2015 IEEE International Conference on Computer Vision (ICCV)*. doi:10.1109/iccv.2015.320
- Wang, X., & Gupta, A. (2016). Generative Image Modeling Using Style and Structure Adversarial Networks. *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 318-335. doi:10.1007/978-3-319-46493-0_20
- Wang, X., Luo, Y., Jiang, T., Gong, H., Luo, S., & Zhang, X. (2011). A New Classification Method for LIDAR Data Based on Unbalanced Support Vector Machine. *2011 International Symposium on Image and Data Fusion*. doi:10.1109/isidf.2011.6024312
- Wang, Z., Bovik, A., Sheikh, H., & Simoncelli, E. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4), 600-612. doi:10.1109/tip.2003.819861
- Weitkamp, C. (2014). *LiDAR*. Springer.
- Weitkamp, C., & Walther, H. (2005). *LIDAR: Range-resolved optical remote sensing of the atmosphere*. Springer.
- Welsh, T., Ashikhmin, M., & Mueller, K. (2002). Transferring color to greyscale images. *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '02*. doi:10.1145/566570.566576

- Wexler, Y., Shechtman, E., & Irani, M. (n.d.). Space-time video completion. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. doi:10.1109/cvpr.2004.1315022
- Wikipedia contributors. (2020a, abril 5). Biharmonic equation. Wikipedia. https://en.wikipedia.org/wiki/Biharmonic_equation
- Wikipedia contributors. (2020b, junio 28). Inpainting. Wikipedia. <https://en.wikipedia.org/wiki/Inpainting>
- Wu, X. (2010). 10 Years of Data Mining Research: Retrospect and Prospect. 2010 IEEE International Conference on Data Mining. doi:10.1109/icdm.2010.172
- Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010). SUN database: Large-scale scene recognition from abbey to zoo. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2010.5539970
- Xie, X., Tian, F., & Seah, H. S. (2007). Feature Guided Texture Synthesis (FGTS) for artistic style transfer. Proceedings of the 2nd International Conference on Digital Interactive Media in Entertainment and Arts - DIMEA '07. doi:10.1145/1306813.1306830
- Xu, L., Yan, Q., & Jia, J. (2013). A sparse control model for image and video editing. ACM Transactions on Graphics, 32(6), 1-10. doi:10.1145/2508363.2508404
- Xu, N., Price, B., Cohen, S., Yang, J., & Huang, T. (2016). Deep Interactive Object Selection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.47

- Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & Dicarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23), 8619-8624. doi:10.1073/pnas.1403112111
- Yan, Z., Zhang, H., Wang, B., Paris, S., & Yu, Y. (2016). Automatic Photo Adjustment Using Deep Neural Networks. *ACM Transactions on Graphics*, 35(2), 1-15. doi:10.1145/2790296
- Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., & Li, H. (2017). High-Resolution Image Inpainting Using Multi-scale Neural Patch Synthesis. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2017.434
- Yeh, R. A., Chen, C., Lim, T. Y., Schwing, A. G., Hasegawa-Johnson, M., & Do, M. N. (2017). Semantic Image Inpainting with Deep Generative Models. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2017.728
- Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. *2017 IEEE International Conference on Computer Vision (ICCV)*. doi:10.1109/iccv.2017.310
- Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., & Sang, N. (2018). BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. *Computer Vision – ECCV 2018 Lecture Notes in Computer Science*, 334-349. doi:10.1007/978-3-030-01261-8_20
- Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative Image Inpainting with Contextual Attention. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. doi:10.1109/cvpr.2018.00577

- Yuan, M. (2008). Geographic Dynamics, Visualization and Modeling. Encyclopedia of GIS, 358-362. doi:10.1007/978-0-387-35973-1_463
- Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful Image Colorization. Computer Vision – ECCV 2016 Lecture Notes in Computer Science, 649-666. doi:10.1007/978-3-319-46487-9_40
- Zhang, Y., David, P., & Gong, B. (2017). Curriculum Domain Adaptation for Semantic Segmentation of Urban Scenes. 2017 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2017.223
- Zhang, Z., Song, Y., & Qi, H. (2017). Age Progression/Regression by Conditional Adversarial Autoencoder. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.463
- Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid Scene Parsing Network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.660
- Zhu, J., Krahenbuhl, P., Shechtman, E., & Efros, A. A. (2015). Learning a Discriminative Model for the Perception of Realism in Composite Images. 2015 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.449
- Zhu, J., Krähenbühl, P., Shechtman, E., & Efros, A. A. (2016). Generative Visual Manipulation on the Natural Image Manifold. Computer Vision – ECCV 2016 Lecture Notes in Computer Science, 597-613. doi:10.1007/978-3-319-46454-1_36
- Zhu, J., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. 2017 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2017.2